الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد – تلمسان–

Université Aboubakr Belkaïd – Tlemcen –

Faculté de Technologie

MEMOIRE

Présenté pour l'obtention du diplôme de MASTER

En : Télécommunications

Spécialité : Télécommunications Optique

Présente Par :

BADAOUI Hamza Ismat

Sujet

Système d'identification sécurisé et de contrôle d'accès par communication optique

Soutenu le 14 juin devant un jury composé de :

FEHAM Mohammed	Professeur Emérite	Université de Tlemcen	Président
BORSALI Ahmed Riad	Professeur	Université de Tlemcen	Encadrant
KARIM Fethallah	Professeur	Université de Tlemcen	Examinateur

Année Universitaire : 2025 - 2026


بسم الله الرحمن الرحيم

Remerciements

Je tiens d'abord à remercier le Tout-Puissant qui m'a donné la volonté, la santé et le courage nécessaires pour faire ce modeste travail.

Je remercie ***Mr BORSALI Ahmed Riad****, mon encadrant. Je le remercie pour son accompagnement, ses conseils, ses idées, son soutien constant, sa disponibilité et la qualité de son encadrement pendant tout ce travail. J'aimerais lui exprimer ici ma reconnaissance et mon respect.*

J'exprime ma plus profonde gratitude à ***Mr FEHAM Mohammed*** *pour l'honneur et la confiance qu'il m'a accordés en acceptant de présider le jury de ce mémoire.*

Je remercie également ***Mr KARIM Fethallah*** *d'avoir accepté d'examiner ce travail.*

Mes sincères remerciements vont à tous les professeurs qui m'ont accompagné durant mon parcours universitaire. La qualité de leur enseignement m'a permis d'acquérir les connaissances nécessaires pour réaliser ce travail.

Je remercie aussi les autres enseignants et toutes les personnes qui ont enrichi ma réflexion par leurs conseils, leurs écrits et nos échanges.

Dédicace

Louange à Dieu toujours et à jamais pour ses bénédictions et ses faveurs.

Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers.

À ma très chère mère, *source de bonheur et de tendresse****,*** *et* ***à mon très cher père,*** *source de sécurité et modèle, tout ce que j'ai accompli, les mots me manquent pour vous dire tout l'amour que je vous porte. Même s'ils existaient, ils ne pourraient jamais exprimer tout ce que je ressens au fond de moi. Je le dois à Dieu et à vous deux. Vous avez toujours été un soutien indéfectible, une source de motivation, d'inspiration et d'encouragement. Vous avez tout sacrifié pour moi. Je prie Dieu de vous protéger de tout mal, de faire de vous une couronne sur ma tête et de vous accorder une longue vie emplie de bonnes actions.*

Mon petit frère, Youcef*, je suis fier d'être ton frère. Pour moi, tu es avant tout mon petit ami. Je prie Dieu de vous protéger, de prendre soin de vous et de vous garder fermes dans sa religion et son obéissance, et puisse-t-il toujours nous aimer les uns les autres, Amen, Seigneur des mondes.*

BADAOUI Hamza Ismat

Sommaire

Introduction générale 1

Chapitre 1

Maison intelligente et domotique

1.1. Introduction 4

1.2. La maison intelligente domotique 4

1.2.1. Historique 4

1.2.2. Qu'est-ce que la domotique ? 6

1.2.3. Définition de la maison intelligente 7

1.2.4. Le principe de la domotique 7

1.2.4.1. Architecture de la maison intelligente 7

1.2.4.2. Les principaux domaines d'application 10

1.3. Technologies utilisées 12

1.3.1. Technologie sans fil (le réseau IoT) 12

1.3.1.1. Bluetooth et BLE 12

1.3.1.2. WI-FI 13

1.3.1.3. ZigBee 13

1.3.1.4. UWB 14

1.3.2. Technologies filaires 14

1.4. Les avantages et les inconvénients de la maison intelligente 15

1.4.1. Avantages 15

1.4.2. Inconvénients 15

1.5. Conclusion 15

Chapitre 2

Architecture matérielle du système

2.1. Introduction 18

2.2. Matériels 18

2.2.1. Arduino 18

2.2.2. ESP32 Wrover Dev avec Camera OV2640 20

2.2.2.1. ESP32 Wrover Dev 20

2.2.2.2. Camera OV2640 22

2.2.3. Ecran LCD (16X2) avec interface I2C 23
2.2.3.1. Ecran LCD (16X2) 23
2.2.3.2. Interface I2C 24
2.2.4. Servomoteur 25
2.2.5. Clavier Matriciels 27
2.2.6. Code QR 28
2.3. Logiciels et applications utilisés 29
2.3.1. Arduino IDE 29
2.3.2. Application MacroDroid 32
2.4. Conclusion 33

Chapitre 3

Réalisation et mise en œuvre d'un système d'accès et de contrôle

3.1. Introduction 35
3.2. Description du projet 35
3.3. Conception et réalisation 35
3.3.1. Programmation les microcontrôleurs 35
3.3.2.1. ESP32 36
3.3.2.2. Arduini Mega 2560 43
3.3.3. Création du code QR 49
3.4. Description de la liaison optique et matérielle 58
3.4.1. Branchage des éléments 60
3.6. Mode de Secours Optique 63
3.6.1. Principe du canal de secours 63
3.6.2. Algorithme de décision et seuillage 63
3.6.3. Liaison et transmission vers l'Arduino Mega 63
3.7. Conclusion 64
Conclusion générale 67
Références bibliographiques 69

Liste des figures

Figure 1.1 Maison intelligente 4
Figure 1.2.ECHO IV 6
Figure 1.3.L'ordinateur de cuisine 6
Figure 1.4. Principe de fonctionnement 6
Figure 1.5. Architecture d'une maison intelligente de 4 niveaux 8
Figure 1.6. Schéma d'un capteur 9
Figure 1.7. Thermostats intelligents. 11
Figure 1.8.Capteur d'énergie intelligent 11
Figure 1.9.Compteur d'énergie intelligent 11
Figure 1.10. Détecteur de la fumée. 12
Figure 1.11. Caméra de surveillance 12
Figure 1.12. Les appareils d'alarmes anti-intrusion 12
Figure 1.13. La relation entre le capteur et l'actionneur 14
Figure 2.1. Carte Arduino Mega 2560 18
Figure 2.2. Les composants de Arduino 19
Figure 2.3. Carte ESP32 Wrover Dev 21
Figure 2.4. Brochage ESP32 Wrover Dev 22
Figure 2.5. Camera OV2640 grand angle 23
Figure 2.6. Ecran LCD 24
Figure 2.7. Interface I2C 25
Figure 2.8. Caractéristiques du I2C. 25
Figure 2.9. Servo Moteur 26
Figure 2.10. Mécanismes du Servo Moteur 26
Figure 2.11. Clavier Matriciels 28
Figure 2.12. Code QR 29
Figure.2.13. Les composants du QR……29
Figure 2.14. Interface du Arduino IDE 30
Figure 2.15. Interface de paramètres de MacroDdroid 32
Figure 3.1. Les étapes pour ajouter les bibliothèques sur Arduino IDE. 36
Figure 3.2. Sélection du type de carte utilisée (ESP32 Wrover kit).. 37
Figure 3.3. Code programme du ESP32. 38
Figure 3.4. Code programme du ESP32. 39

Figure 3.5. Code programme du ESP32. 40
Figure 3.6. Code programme du ESP32. 41
Figure 3.7. Code programme du ESP32. 42
Figure 3.8. Préparation du ESP32.. 42
Figure 3.9. Déterminer le type de la carte. 43
Figure 3.10. Code de programmation du Arduino Mega. 44
Figure 3.11. Code de programmation du Arduino Mega. 45
Figure 3.12. Code de programmation du Arduino Mega. 46
Figure 3.13. Code de programmation du Arduino Mega. 47
Figure 3.14. Code de programmation du Arduino Mega. 48
Figure 3.15. Affichage du moniteur de l'Arduino. 49
Figure 3.16. Création du code QR. 49
Figure 3.17. Accès à l'interface de création d'une nouvelle macro. 50
Figure 3.18. Sélection et configuration du déclencheur d'événement. 51
Figure 3.19. Définition du mode d'interaction via le bouton flottant.. 52
Figure 3.20. Personnalisation de l'icône de commande pour l'interface utilisateur. 53
Figure 3.21. Sélection de la catégorie d'action à associer au bouton. 54
Figure 3.22. Configuration du chemin d'accès vers le stockage local. 55
Figure 3.23. Filtrage et sélection de l'image du code QR source. 56
Figure 3.24. Validation finale. 57
Figure 3.25. Interface utilisateur finale. 57
Figure 3.26. Code QR utilisé……..58
Figure 3.27. Le succès décodage du code QR. 58
Figure 3.28. Montage de système. 59
Figure 3.29. Montage en réel. 60
Figure 3.30. Maquette du système. 62
Figure 3.31. Partie du programme relative au mode de secours. 64
Figure 3.32. Partie du programme relative au mode de secours 64

Liste des tableaux

Tableau 2.1. Les différents composants de l'Arduino 19
Tableau 2.2. Fiches techniques du Arduino Mega2560. 20
Tableau 3.1. Composants du montage. 59
Tableau 3.2. Branchage des composants. 60

Résumé :

Ce mémoire présente la conception et la réalisation d'un système de contrôle d'accès sécurisé pour une maison intelligente, s'appuyant sur l'utilisation de QR codes et de codes PIN pour automatiser l'ouverture des portes. Après avoir présenté les fondements de la domotique et des protocoles de communication sans fil comme le Wi-Fi et le Bluetooth, ce travail s'intéresse à la mise en œuvre concrète utilisant une carte Arduino Mega, un module ESP32-CAM pour la reconnaissance optique et un clavier numérique. L'objectif est de démontrer comment l'intégration de technologies électroniques abordables peut renforcer la sécurité domestique tout en offrant une gestion flexible et moderne de l'habitat.

Mot clé : Maison intelligente, domotique, Arduino, ESP32, camera ov2640, capteurs, code QR, communication optique, sécurité.

Abstract:

This thesis presents the design and implementation of a secure access control system for the smart home, using QR codes and PIN codes to automate door opening. After exploring the fundamentals of home automation and wireless communication protocols such as Wi-Fi and Bluetooth, the work details a practical implementation using an Arduino Mega board, an ESP32-CAM module for optical character recognition, and a keypad. The objective is to demonstrate how the integration of affordable electronic technologies can enhance home security while providing flexible and modern home management.

Key words: Smart home, home automation, Arduino, ESP32, OV2640 camera, sensors, QR code, optical communication, security.

الملخص

تقدم هذه الأطروحة تصميم وتنفيذ نظام تحكم آمن في الوصول إلى المنزل الذكي، باستخدام رموز الاستجابة السريعة (QR) ورموز التعريف الشخصية (PIN) لأتمتة فتح الأبواب. بعد استعراض أساسيات أتمتة المنازل وبروتوكولات الاتصال اللاسلكي مثل الواي فاي والبلوتوث، تفصل الأطروحة تطبيقا عمليا باستخدام لوحة أردوينو ميجا، ووحدة ESP32-CAM للتعرف الضوئي على الأحرف، ولوحة مفاتيح. يهدف هذا البحث إلى توضيح كيف يمكن لدمج التقنيات الإلكترونية ذات التكلفة المعقولة أن يعزز أمن المنزل مع توفير إدارة منزلية مرنة وعصرية.

الكلمات المفتاحية :المنزل الذكي، الأتمتة المنزلية، أردوينو، ESP32، كاميرا OV2640، أجهزة الاستشعار، رمز الاستجابة السريعة(QR) ، الاتصال البصري، الأمن.

Introduction générale

Introduction générale

De nos jours, l'évolution rapide des technologies électroniques, informatiques et des réseaux de communication a profondément modifié notre approche de l'habitat. La domotique, longtemps perçue comme une innovation futuriste aux usages restreints, s'est aujourd'hui imposée comme une solution concrète et accessible, s'intégrant progressivement aux habitations modernes. Elle vise principalement à améliorer le confort des résidents, à renforcer la sécurité des biens et des personnes, et à optimiser la consommation énergétique des édifices. Le principe d'une habitation intelligente repose sur l'interconnexion d'objets capables de dialoguer entre eux. Ceci permet d'automatiser diverses tâches journalières ou d'offrir un contrôle à distance via un smartphone, une tablette ou un ordinateur.

La sécurité constitue un volet essentiel parmi les applications importantes de la domotique. Les systèmes de verrouillage conventionnels présentent en effet des vulnérabilités, telles que le risque de perte de clés, les tentatives d'effraction ou la reproduction illicite. Il devient donc impératif d'intégrer des technologies modernes d'identification et d'authentification pour garantir un niveau de sécurité supérieur. C'est dans ce cadre que notre projet vise à concevoir et à développer un système intelligent de contrôle d'accès reposant sur l'association d'un code QR et d'un code PIN. Cette approche de double authentification permet de renforcer la sécurité, tout en proposant une solution simple, rapide et économique, particulièrement adaptée aux maisons intelligentes.

Le système que nous avons élaboré s'appuie principalement sur une carte Arduino Mega 2560, conjointement à un module ESP32-CAM qui assure la lecture et l'analyse des codes QR. Divers composants électroniques complètent l'architecture du système : un écran LCD pour l'affichage des données, un clavier matriciel pour la saisie du code PIN, un servomoteur simulant l'ouverture de la porte, et une LED RGB signalant l'état opérationnel. L'environnement de développement Arduino IDE nous a permis de programmer les divers modules, garantissant ainsi la communication inter-composants et l'opérabilité du système.

Ce projet a été l'occasion d'appliquer concrètement les connaissances acquises dans les domaines des systèmes embarqués, de la programmation des microcontrôleurs et des technologies de communication propres aux habitations intelligentes. Il représente également une expérience formatrice, offrant une compréhension approfondie des phases de conception, de programmation et de mise en œuvre d'un système domotique sécurisé.

Ce mémoire s'organise en trois grandes sections distinctes :

Le premier chapitre expose les concepts essentiels d'une maison intelligente et de la domotique, en retraçant leur historique, leurs principes de fonctionnement, leurs applications et les technologies qu'elles emploient.

Le deuxième chapitre décrit les équipements matériels et logiciels mis en œuvre pour ce projet. Nous y détaillons les divers composants électroniques, comme les cartes Arduino, le module ESP32-CAM, les capteurs et les périphériques, de même que les outils logiciels employés pour programmer et simuler le système.

Enfin, le troisième chapitre examine en détail la conception et l'implémentation concrète du système de contrôle d'accès sécurisé. Nous y précisons le montage électronique, les connexions effectuées, la programmation des microcontrôleurs et les tests réalisés pour valider son fonctionnement correct.

Chapitre 1

Maison intelligente et domotique

1.1. Introduction :

Qui d'entre nous ne rêve pas de posséder une belle maison intelligente, dotée de toutes les technologies électroniques modernes qui l'aident à accomplir ses tâches quotidiennes et lui procurent confort et sécurité ? C'est ce dont nous allons parler dans ce chapitre, et nous verrons ce qui distingue la maison intelligente des autres, En considérant ses atouts majeurs, les inconvénients potentiels, ainsi que ses principaux champs d'application et les matériaux requis (figure.1.1).

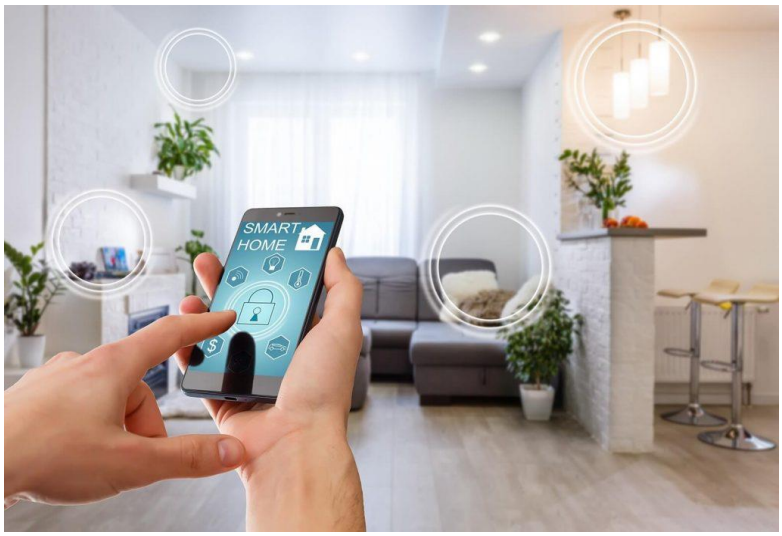


Figure1.1 : Maison intelligente. **[1]**

1.2. La maison intelligente domotique :

1.2.1. Historique :

La domotique fit son apparition en 1966 avec la mise en vente du premier appareil capable de gérer plusieurs tâches ménagères de manière autonome : le ECHO IV, un ordinateur de la taille d'une salle à manger. Il avait la capacité de contrôler divers éléments tels que la télévision, la température de la maison ou encore le réveil matin. Bien que considérée comme un échec en raison de son encombrement et de sa complexité d'utilisation, cette technologie était en avance sur son temps et a jeté les bases de l'installation domotique moderne. **[2]**

L'automatisation des maisons a émergé dans les années 1970 avec la miniaturisation des systèmes électroniques et informatiques. Le développement d'appareils électriques s'est étendu aux objets de tous les jours, diminuant la consommation d'énergie et améliorant la performance des appareils. **[3]**

Les Grandes Étapes de l'Histoire de la Domotique :

- Le début du XXe siècle (1900-1920) a vu l'arrivée de petits appareils ménagers motorisés, comme les aspirateurs et les machines à laver. Ce fut le point de départ de l'allègement des corvées ménagères.
- Les années 1960 furent une période d'essais : ECHO IV (**E**lectronic **C**omputing **H**ome **O**perator) en anglais (figure 1.2), est souvent vu comme le premier ordinateur personnel (voir figure 1.2). Il pouvait gérer la température de la maison et enregistrer des listes de courses, mais il n'a pas été vendu au public.
- L'ordinateur de cuisine (1969) était une machine Honeywell **(**figure 1.3) conçue pour classer des recettes, mais elle était trop compliquée pour son temps.
- 1975 représente un tournant, le protocole X10 a permis, pour la première fois aux appareils de communiquer via le réseau électrique (courant porteur), sans nécessiter de travaux.
- Dans les années 1980 et 1990, le concept s'est popularisé. L'expression Smart House a été officiellement adoptée en 1984 et l'intégration du câblage a commencé dès la construction des habitations.
- Les années 2000, avec l'Internet des Objets (IoT), ont transformé la domotique grâce au Wi-Fi et aux smartphones. Le contrôle, initialement local, est devenu possible à distance, partout dans le monde. **[4]**

Initialement, l'automatisation visait à procurer plus de confort, de sécurité et de facilité dans la gestion des habitations, en utilisant l'intégration et la multifonctionnalité. Ces deux aspects devaient permettre un fonctionnement autonome tout en étant liés aux autres appareils domestiques (intégration), réduisant ainsi la consommation d'énergie, et améliorant la sécurité et la communication (multifonctionnalité).

Pour cette raison, les consommateurs sont de plus en plus attirés par l'automatisation, en particulier par les appareils connectés **[3].**



Figure1.2. ECHO IV **[5].**



Figure.1.3. L'ordinateur de cuisine [**6].**

1.2.2. Qu'est-ce que la domotique ? :

La domotique combine le mot latin (**domus)** qui veut dire maison, avec le suffixe (**tique**), souvent employé pour parler des technologies, également appelée maison connectée, maison intelligente ou encore (smart home), est de mettre en réseau tous les objets connectés de la maison. C'est l'ensemble des technologies électroniques, d'automatisation, informatiques et de télécommunications utilisées dans les bâtiments et les maisons. La domotique vise principalement à améliorer le confort, la sécurité et la gestion de l'énergie, simplifiant ainsi la vie quotidienne des habitants (figure 1.4) [**7].**



Figure.1.4. Principe de fonctionnement **[8].**

1.2.3. Définition de la maison intelligente :

En termes simples, une maison intelligente relie plusieurs appareils via un système de communication central. Ce dispositif, accessible sur votre téléphone ou tablette, permet de gérer à distance ou sur place divers éléments comme les serrures, les caméras, ou les systèmes de divertissement, soit directement, soit via une programmation horaire basée sur des déclencheurs spécifiques **[9].**

1.2.4. Le principe de la domotique :

Aujourd'hui, les objets connectés dans nos maisons vont au-delà de l'automatisation et du contrôle à distance, ils fonctionnent ensemble afin d'améliorer le confort des habitants, d'accroître la sécurité, et d'ajuster au mieux l'énergie consommée.

La domotique implique de connecter divers appareils dans une maison et de centraliser leur gestion. Il s'agit souvent d'appareils courants comme (éclairage, chauffage, ventilation, production solaire, alarme, volets roulants, porte de garage, écrans multimédias…) auxquels on ajoute des moyens de communication. Chaque appareil est lié à d'autres via un processus d'association, qui permet, par exemple, à un interrupteur de savoir quelles lampes allumer. Cette association peut se faire directement entre deux objets, ou par le biais d'un boîtier domotique.

Chaque ensemble d'appareils est contrôlable via une application sur smartphone, tablette, ordinateur ou télécommande, cette application permet de piloter à distance des instructions, comme augmenter la température, allumer une lumière, ou démarrer la télévision.

Ces objets sont considérés comme intelligents. Ils sont équipés de capteurs, par exemple des capteurs de température ou de présence pour un thermostat, qui mesurent et détectent les habitudes des occupants. Des informations comme les allées et venues, ou le temps passé dans une pièce, sont enregistrées et transmises aux radiateurs pour adapter la température en fonction des programmes établis. Ainsi, le chauffage ne fonctionne que lorsque cela est nécessaire.

L'emploi d'appareils connectés, comme des caméras et des alarmes, contribue aussi à renforcer la sûreté des personnes et la sécurité des biens, plus spécifiquement pour les enfants et les aînés et les handicapèes**. [7]**

1.2.4.1. Architecture de la maison intelligente :

Les maisons intelligentes intègrent divers capteurs et actionneurs qui surveillent l'environnement, l'activité et les fonctions physiologiques, le tout relié par un réseau sans fil.

Les progrès dans les systèmes de détection, d'actionnement et de communication à faible consommation, combinés aux technologies de communication modernes et aux plateformes informatiques abordables comme les FPGA, les microcontrôleurs et les microprocesseurs, ont permis de créer des systèmes de maison intelligente peu coûteux (figure 1.5) **[23].**

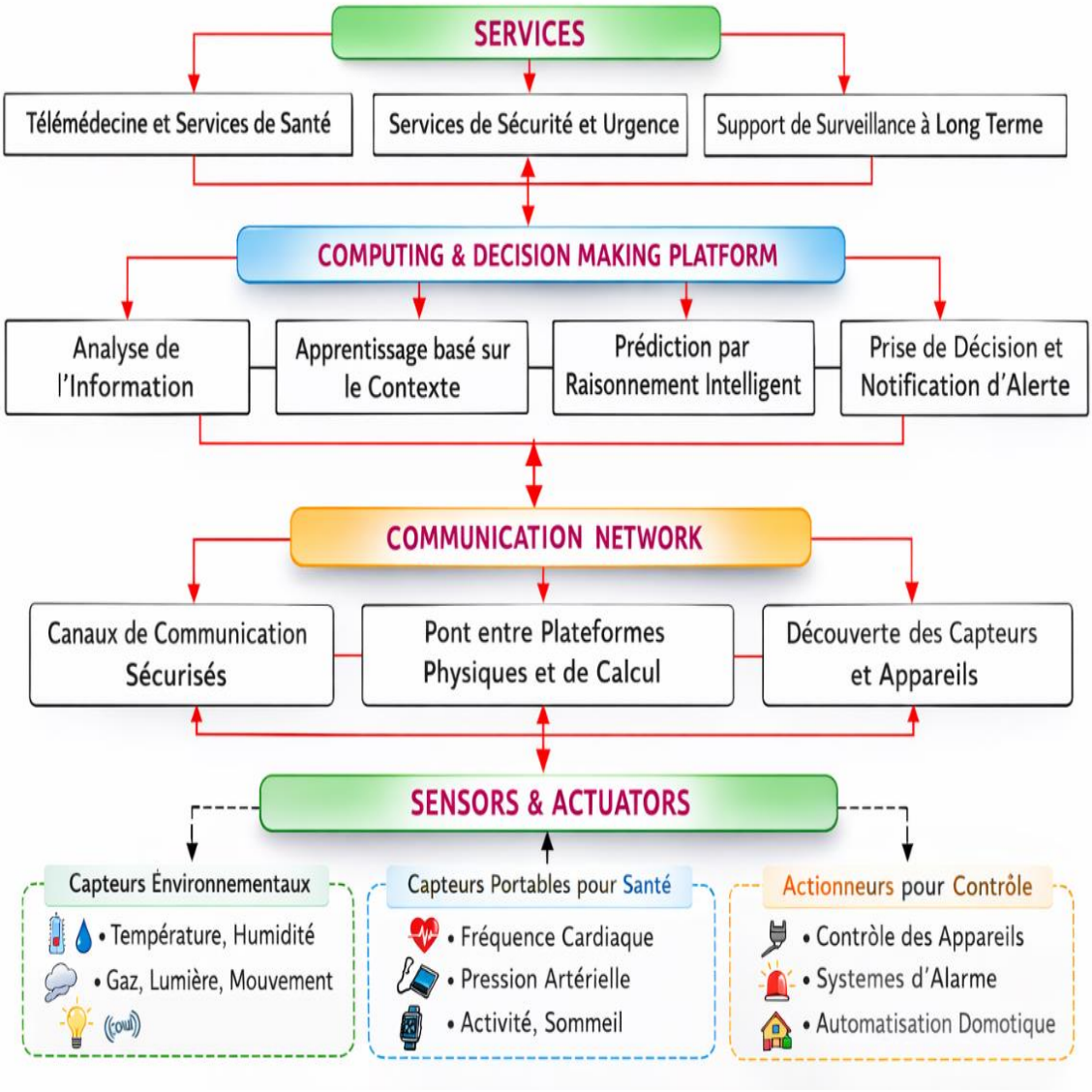


Figure.1.5. Architecture d'une maison intelligente de 4 niveaux **[18].**

1.2.4.1.1. Les capteurs :

Dans les habitations intelligentes, les capteurs et actionneurs servent de pont entre le monde physique et le numérique. Ils transforment une information physique en un signal électrique (la figure 1.5).Les capteurs recueillent diverses informations environnementales et physiologiques, allant des conditions de la maison à la santé des habitants. Les actionneurs réagissent à ces informations en ajustant l'environnement ou en prenant des mesures

spécifiques, le tout grâce à une communication sans fil avec une plateforme centrale (figure 1.6) **[24].**

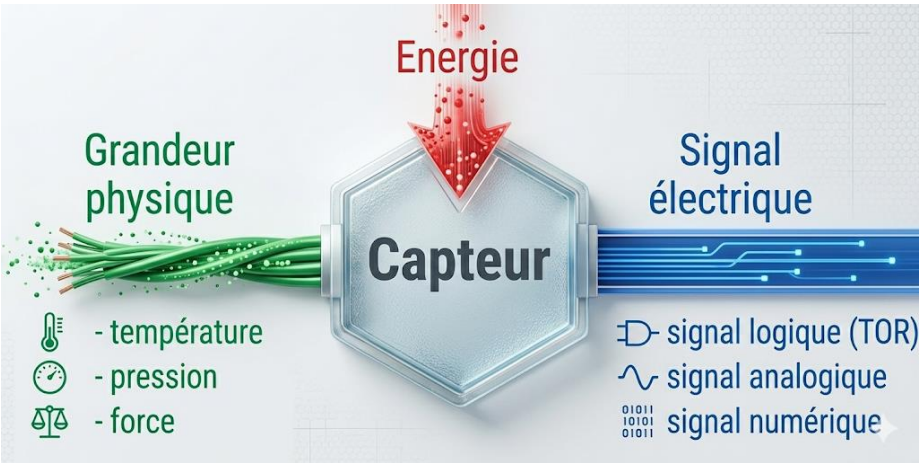


Figure.1.6. Schéma d'un capteur **[20]**.

1.2.4.1.2. Réseau de communication :

Dans une maison intelligente, les capteurs et actionneurs transmettent les données physiologiques et environnementales à une plateforme centrale via un réseau de communication, formant la deuxième couche de l'architecture. Si la connexion filaire convient aux capteurs fixes, elle est inadaptée aux systèmes portables car elle limite la mobilité et risque des déconnexions. Les textiles conducteurs constituent une alternative, mais leur faible résistance au lavage compromet la connectivité à long terme. Ainsi, les technologies sans fil à faible consommation s'imposent comme la solution la plus fiable pour assurer les communications à courte portée dans cet environnement **[19].**

1.2.4.1.3. Plateforme informatique et décisionnelle:

Le contrôleur central sert de cerveau. Il utilise des appareils tels que les smartphones ou les ordinateurs pour recueillir des informations provenant de capteurs et d'actionneurs. Ces informations sont ensuite analysées pour informer l'utilisateur et piloter les actionneurs. La plateforme conserve aussi ces données, présente les résultats et met en œuvre des algorithmes prédictifs basés sur l'intelligence artificielle et l'apprentissage automatique afin de comprendre l'environnement de la maison et les habitudes de ses occupants **[19].**

1.2.4.1.4. Services :

Dans l'architecture d'une maison intelligente, les fournisseurs de services offrent directement aux résidents des prestations liées à la santé, à l'environnement, à la sécurité et à

la protection. La plateforme de passerelle joue un rôle central en tant que principal fournisseur de ces services. Elle active les actionneurs pour gérer l'environnement domestique et se sert de l'intelligence artificielle pour évaluer la sécurité et l'état de l'habitation, offrir des services adaptés aux besoins des occupants, et surveiller les situations potentiellement dangereuses **[19]**.

1.2.4.2. Les principaux domaines d'application :

1.2.4.2.1. Le confort :

L'automatisation résidentielle améliore le confort à la maison. On peut contrôler facilement l'éclairage, le chauffage, des points lumineux de la TV et les volets roulants. Avec une application sur leur téléphone, les habitants peuvent régler l'ouverture des volets et la température selon l'heure. Des détecteurs répartis dans la maison repèrent les personnes et peuvent allumer ou éteindre les lumières, ajuster la température ou même lancer de la musique d'ambiance si les habitants le souhaitent.**[7]**

1.2.4.2.2. L'énergie :

Le secteur est crucial, car beaucoup de gens s'inquiètent du prix de l'électricité et du gaz. Un but de la domotique est d'améliorer l'efficacité énergétique des bâtiments et de diminuer la consommation d'eau en repérant les fuites. Divers systèmes automatisés, comme la ventilation et la gestion du chauffage, adaptent les maisons aux conditions climatiques extérieures et aux habitudes intérieures. L'objectif est de diminuer les dépenses énergétiques tout en gardant le confort des habitants **[7],** en utilisent des appareilles électroniques comme (Thermostats intelligents, capteur de température et d'humidité et compteur d'énergie intelligent), (figures 1.7, 1.8 et 1.9).



Figure 1.7. Thermostats intelligents **[11]**.



Figure 1.8. Capteur d'énergie intelligent **[12]**.



Figure.1.9. Compteur d'énergie intelligent **[13].**

1.2.4.2.3. La sécurité :

Dans un bâtiment, les capteurs domotiques peuvent détecter les mouvements, identifier la présence d'une personne ou signaler la fumée. Ces informations contribuent à améliorer la sécurité des habitations et de leurs occupants face aux vols et aux accidents qui peuvent survenir à la maison **[7].** On parle alors d'une maison bienveillante, en trouve des équipements électronique et informatique pour cette sécurise comme par exemple (Caméra de surveillance,

détection de mouvement, détection de fumé et de gaz, serrures électroniques et control d'accès (RFID)) (figure 1.12, 1.13 et 1.14).



Figure.1.10. Détecteur de la fumée **[14].**



Figure.1.11. Les appareils d'alarmes anti-intrusion **[15]**.



Figure.1.12. Caméra de surveillance **[16].**

1.3. Technologies utilisées :

L'essor des nouvelles technologies a engendré une pléthore de possibilités pour les objets connectés. La domotique, qui repose sur la mise en réseau d'appareils électriques dans un bâtiment, représente un des marchés les plus intéressants. Ces appareils peuvent être aisément commandés par une unité centrale programmable. Les plus importants d'entre eux :

- La communication sans filaire.
- La communication filaire.

1.3.1. Technologie sans fil (le réseau IoT) :

Les technologies de communication sans fil à courte portée permettent une transmission sans fil sur une distance limitée, généralement de quelques dizaines à quelques centaines de mètres. (Le Bluetooth, le Wi-Fi et le Zigbee… ect) sont des exemples courants **[17]**.

1.3.1.1. Bluetooth et BLE :

Le Bluetooth est une technologie sans fil répandue pour les courtes distances. On la trouve partout, des écouteurs aux systèmes embarqués, en passant par les montres connectées et les moniteurs d'activité.

La version 6.0 du Bluetooth, parue en septembre 2024, a introduit des fonctions nouvelles, comme la détection de canal. Pourtant, les versions 4.0, 5.0 et supérieures sont les

plus utilisées. La version 5.0 propose des vitesses de transmission atteignant 2 Mbit/s, alors que la 4.2 monte à 1 Mbit/s **[17].**

Pour régler le problème de consommation d'énergie du Bluetooth, le Bluetooth Low Energy (BLE) a été créé. Il faut noter que le BLE est fait pour les appareils basse consommation de l'internet des objets. Il ne remplace pas le Bluetooth classique. Le BLE se sert de la même bande ISM 2.4 GHz que le Bluetooth. Au contraire du Bluetooth classique, qui accepte jusqu'à sept appareils connectés à un seul appareil maître, le BLE en autorise jusqu'à 128. Le BLE utilise 40 canaux de 2 MHz et une méthode de saut de fréquence pour améliorer les performances et réduire les interférences **[17].**

1.3.1.2. WI-FI :

Le Wi-Fi constitue 31 % des liens IoT. Nous discuterons ici du Wi-Fi normal, ainsi que du Wi-Fi HaLow (802.11ah), fait pour les applications IoT à longue portée et à faible consommation d'énergie.

Le WiFi 6 marche sur les mêmes fréquences 2.4 GHz et 5 GHz que les autres systèmes sans fil. La version 6E accepte aussi la fréquence 6 GHz. La distance couverte change beaucoup, allant de 10 m à l'intérieur à plus de 100 m à l'extérieur, selon les conditions et la puissance du signal.

Le WiFi 6/6E (802.11ax), sorti en 2021, est ce qui se fait de plus courant aujourd'hui. En termes de vitesse, il peut aller jusqu'à 9.6 Gbit/s, ce qui est plus rapide que le WiFi 5 (802.11ac), qui atteignait 3.5 Gbit/s.

Le WiFi HaLow fonctionne en dessous de 1 GHz. Il traverse mieux les murs et porte plus loin (jusqu'à 1 km) tout en utilisant moins d'énergie.

1.3.1.3. ZigBee :

ZigBee est une norme de communication sans fil peu coûteuse et économe en énergie, conçue pour les réseaux personnels et particulièrement adaptée aux applications industrielles et domotiques. Bien que moins courant que le Wi-Fi, ZigBee gagne en popularité dans les maisons intelligentes, notamment pour les ampoules, les thermostats et les capteurs de sécurité.

Ce protocole a été créé en 2002 avec la fondation de la ZigBee Alliance (désormais Connectivity Standards Alliance). De nos jours, des entreprises importantes telles que Philips, Texas Instruments, Samsung et Amazon participent à son perfectionnement.

En réalité, ZigBee a été pensé pour l'automatisation, offrant une configuration et une connectivité simples des appareils, une faible consommation d'énergie qui prolonge la durée de vie des batteries, et une sécurité renforcée.

1.3.1.4. UWB :

L'UWB (ultra-large bande) est un protocole de communication récent qui gagne en popularité. Bien qu'il ne soit pas encore très répandu, on le trouve de plus en plus dans nos technologies, des téléphones aux voitures, en passant par les maisons connectées et l'industrie.

Comme d'autres technologies sans fil, l'UWB utilise un spectre de fréquences spécifique. Mais, contrairement aux systèmes à bande étroite, elle émet sur une large bande passante allant de 3.1 GHz à 10.6 GHz. Elle a une portée habituelle de 1 à 50 mètres et fonctionne mieux quand il n'y a pas d'obstacles entre les appareils.

La puissance maximale d'émission de l'UWB est de 41.3 dBm/MHz, soit environ 0.5 mW en moyenne. Cette faible puissance réduit les interférences avec les systèmes existants comme le Wi-Fi ou le Bluetooth. De plus, elle renforce la sécurité de l'UWB, car les signaux sont plus difficiles à détecter en raison de leur large plage de fréquences et de leur faible puissance.

1.3.2. Technologies filaires :

Dans le domaine de la construction et de la rénovation immobilière, le BUS est un câblage qui assure la liaison entre les commandes et le tableau électrique, lequel intègre les modules domotiques, via un serveur qui constitue le centre de contrôle du système **[18]**. Grâce à cette communication bidirectionnelle, les unités d'entrée transmettent des informations aux récepteurs de sortie, lesquels gèrent les équipements électriques comme l'éclairage et les ouvertures. De plus, les récepteurs rendent compte de leur état aux unités d'entrée. L'installation se compose de deux réseaux : un réseau bus filaire connectant les capteurs aux actionneurs et un réseau d'alimentation reliant ces derniers au courant fort (figure 1.13) **[19].**

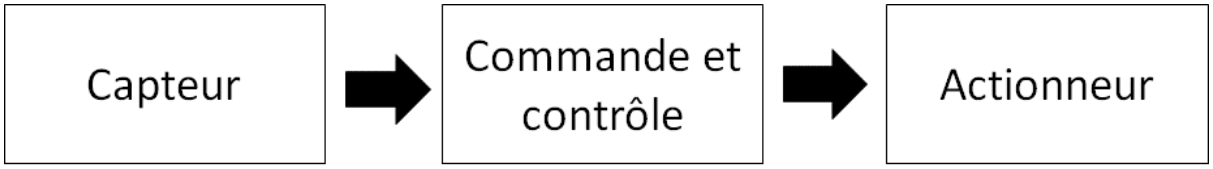


Figure1.13. La relation entre le capteur et l'actionneur **[19].**

Le choix se résume à 3 types de câbles principaux :

- Bien que les bus UTP soient courants, il est conseillé de privilégier les catégories supérieures (5 ou 6) pour une performance accrue.
- Les bus STP et FTP, grâce à leur blindage, offrent une qualité supérieure comparée aux UTP.

- Les câbles VDI (voix, données, images) permettent l'intégration de fonctions audiovisuelles, ce qui n'est pas possible avec les autres types de câbles.

1.4. Les avantages et les inconvénients de la maison intelligente :

1.4.1. Avantages :

- Les maisons intelligentes peuvent améliorer la qualité de vie grâce à des services comme la détection d'incendie, la surveillance de la qualité de l'air et les systèmes de sécurité.
- Elles donnent accès à des options pratiques comme l'éclairage automatique, les serrures sans clé et la programmation à distance.
- Les technologies de la maison intelligente peuvent laisser plus de temps pour les loisirs.
- Elles peuvent aider à économiser de l'argent en réduisant la consommation d'énergie et les coûts de fonctionnement.
- Elles peuvent aussi aider les personnes âgées à vivre de façon plus autonome et sécurisée **[18]**.

1.4.2. Inconvénients :

- Sécurité : Tout appareil connecté à Internet peut avoir des failles de sécurité. Il y a un risque qu'une personne puisse pirater votre système de maison intelligente, ce qui pourrait exposer votre vie privée. Un cambrioleur pourrait aussi surveiller votre système d'alarme ou votre caméra de sécurité.
- Dépendance au fabricant : Si vos appareils ne fonctionnent qu'avec les serveurs du fabricant, cela peut poser problème si l'entreprise fait faillite. Il est important que les appareils de maison intelligente soient compatibles entre eux pour fonctionner correctement.
- Coûts : Pour transformer une habitation standard en maison intelligente, il est important de prendre en considération les dépenses liées à l'acquisition et à la mise en place des équipements requis. Bien que l'investissement initial puisse être conséquent, il est possible d'anticiper une réduction des coûts énergétiques à long terme **[18]**.

1.5. Conclusion :

Dans ce chapitre, nous avons examiné les points essentiels de la domotique, y compris sa définition, son histoire et ses applications dans des secteurs comme la sécurité et la santé dans les maisons intelligentes, sans oublier ses atouts et ses limites. L'acquisition de cette technologie peut être envisagée par ceux qui en ont les moyens, car elle simplifie la vie de tous

les jours, surtout pour les enfants et les personnes âgées. Le prochain chapitre portera sur les outils employés pour mettre en place un système d'identification sécurisé.

Chapitre 2

Architecture matérielle du système

2.1. Introduction :

Le présent chapitre s'attache à exposer les principaux constituants, qu'ils soient matériels ou logiciels, mobilisés pour la conception et la mise en œuvre de notre système d'identification et de contrôle d'accès sécurisé. Chaque composant sera par la suite décrit en détail, afin d'en comprendre la fonction spécifique et la contribution essentielle à l'assurance d'un fonctionnement optimal du système.

2.2. Matériels :

2.2.1. Arduino :

La carte Arduino est un circuit imprimé conçu pour accueillir un microcontrôleur et permettre l'accès à toutes ses entrées et sorties. Elle intègre aussi quelques composants électroniques qui servent à faire fonctionner le microcontrôleur ou à élargir ses possibilités.

Un microcontrôleur est un petit ordinateur contenu dans une seule puce. Il constitue une solution pratique pour programmer et contrôler des dispositifs électroniques. Il existe de nombreux types de cartes avec microcontrôleur, parmi lesquelles on trouve la platine Wiring, le PIC, le Basic Stamp, et bien sûr Arduino **[21].**

L'Arduino Mega 2560 ressemble au Due par son organisation des pins et sa compatibilité avec les shields de l'UNO. Il propose 54 entrées/sorties numériques, 16 entrées analogiques et 15 sorties PWM. Contrairement au Due qui utilise un microcontrôleur ARM, il est équipé d'un ATMEGA 2560. Sa fréquence de fonctionnement est de 16 MHz. Il offre 256 KB de mémoire ROM, 8 KB de RAM, 4 KB d'EEPROM et fonctionne sous une tension de 5V (figure 2.1) **[41].**

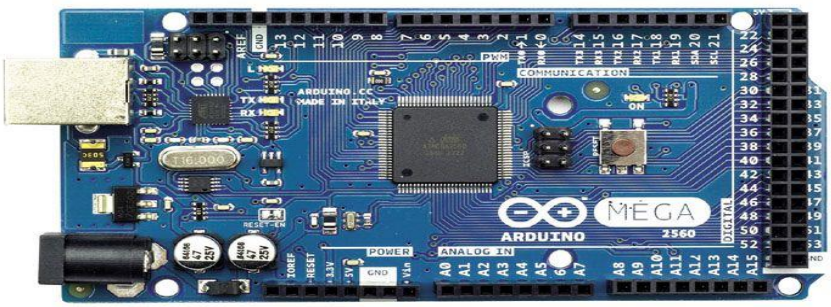


Figure.2.1. Carte Arduino Mega **[23].**

La carte Arduino contient plusieurs éléments essentiels, notamment (figure.2.2**)** :

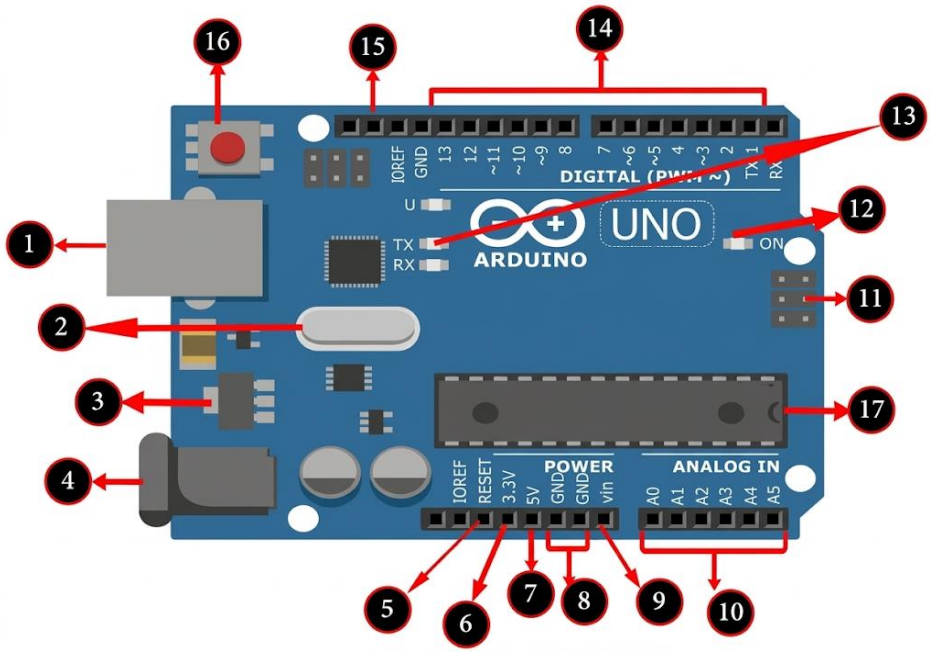


Figure.2.2. Les composants de Arduino **[22].**

1	PowerUSB
2	Crystal Oscillator
3	Régulateur de tension
4	Power (Barrel Jack)
5	Arduino Reset
6	Pin (3.3v) - Sortie alimentation
7	Pin (5) - Sortie alimentation
8	Pin (GND)
9	Pin (Vin)
10	Analog pins
11	Pin ICSP
12	Indicateur d'alimentation LED
13	LEDs TX et RX
14	E/S numérique
15	AREF
16	Arduino Reset
17	Microcontrôleur

Tableau.2.1. Les différents composants de l'Arduino **[22].**

Les caractéristiques en détail du Arduino Méga 2560 sont présentées dans le tableau ci-dessous **(**tableau.2.2**) :**

Paramètre	Détails Techniques
Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limite)	6-20V
Broches d'E/S numériques	54 (dont 15 peuvent fournir une sortie PWM)
Broches d'entrée analogique	16
Courant continu par broche E/S	20 mA
Courant pour la broche 3.3V	50 mA
Mémoire Flash	256 Ko (dont 8 Ko utilisés par le bootloader)
SRAM	8 Ko
EEPROM	4 Ko
Vitesse d'horloge	16 MHz
LED intégrée	Connectée à la broche 13

Tableau 2.2. Fiches techniques du Arduino Mega2560 **[42].**

2.2.2. ESP32 Wrover Dev avec Camera OV2640:

2.2.2.1. ESP32 Wrover Dev :

L'ESP32-WROVER est un module microcontrôleur WiFi, Bluetooth et BLE polyvalent, adapté à de nombreuses applications. Il peut aller des réseaux de capteurs qui consomment peu d'énergie jusqu'à des tâches plus lourdes comme l'encodage vocal, la lecture en continu de musique ou le décodage MP3.

Ce module repose sur la puce ESP32-D0WDQ6, la même que celle utilisée dans le module ESP-WROOM-32. Cependant, l'ESP32-WROVER dispose en plus d'une mémoire vive pseudo-statique (PSRAM) SPI de 32 Mbits. Il inclut 4 Mo de mémoire flash SPI externe ainsi que 4 Mo de PSRAM externe.

Le module ESP32-WROVER est muni d'une antenne PCB, tandis que la version ESP32-WROVER-I utilise une antenne IPEX. Les informations de cette fiche technique concernent les deux versions.

La puce à l'intérieur est conçue pour être flexible et évolutive. Elle contient deux cœurs de processeur qui peuvent être contrôlés séparément, avec une fréquence d'horloge réglable entre 80 MHz et 240 MHz. Il est aussi possible d'éteindre le processeur principal et d'utiliser un coprocesseur basse consommation. Ce dernier peut alors surveiller les périphériques en continu et détecter des changements ou des seuils franchis. Par ailleurs, l'ESP32 intègre différents types de périphériques, comme des capteurs tactiles capacitifs, des capteurs à effet Hall, des amplificateurs à faible bruit, ainsi que des interfaces pour carte SD, Ethernet, SPI rapide, UART, I2S et I2C **(**figure 2.3**) [42].**

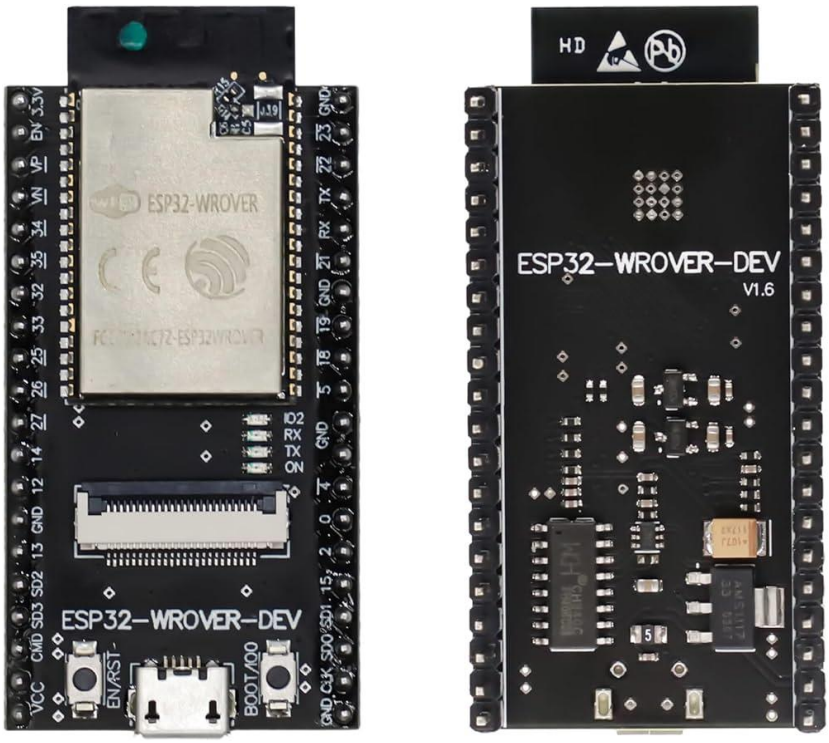


Figure.2.3. Carte ESP32 Wrover Dev **[31].**

Le schéma suivant **(**figure 2.4), illustre le brochage de la carte ESP32-Wrover CAM.

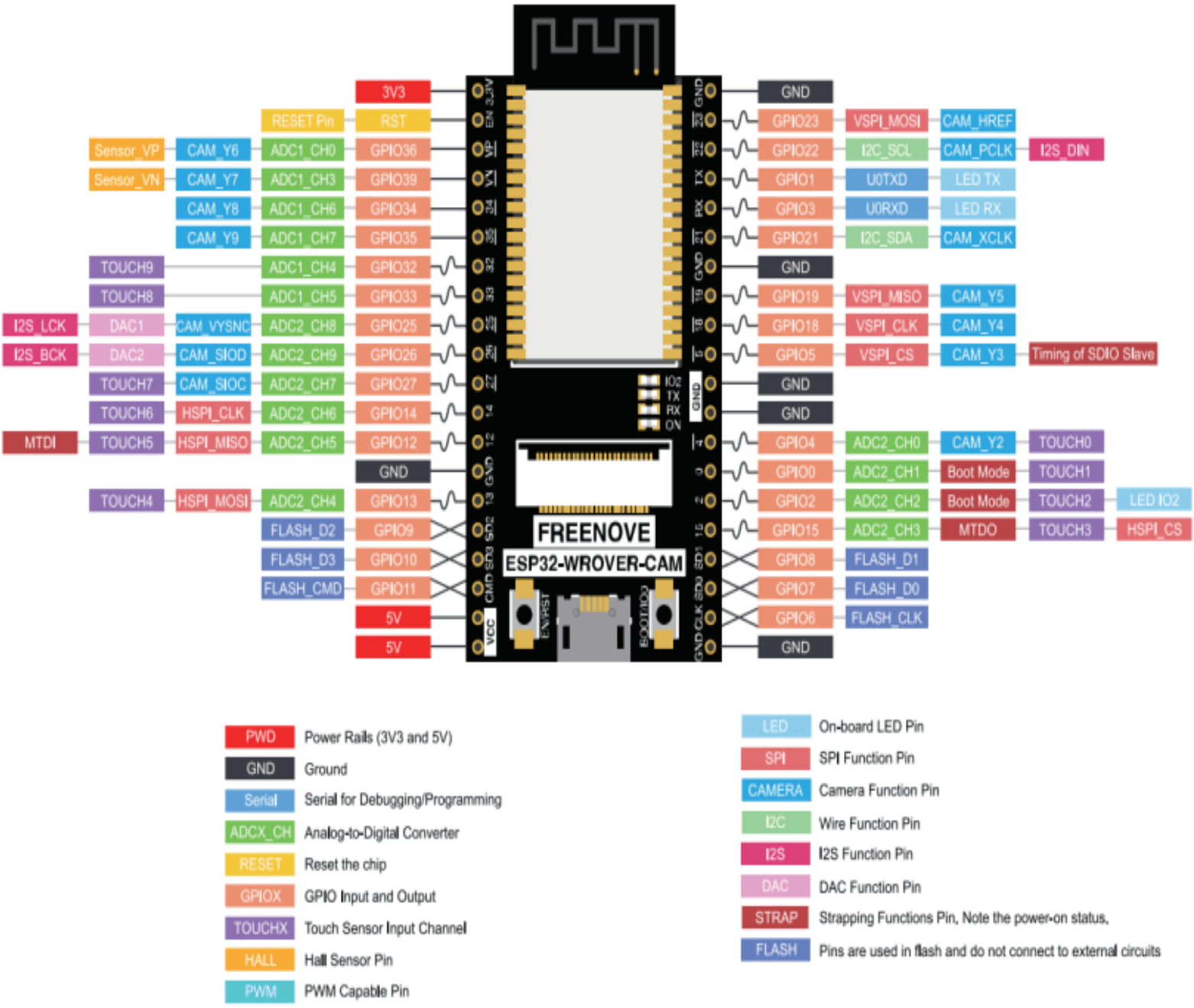


Figure.2.4. Brochage ESP32 Wrover Dev **[24]**.

2.2.2.2. Camera OV2640:

L'OV2640 est un capteur d'image CMOS UXGA de 1/4 pouce avec une résolution de 1632 x 1232 pixels.

Sa taille compacte et sa faible tension d'alimentation lui permettent d'intégrer les fonctions d'une caméra UXGA ainsi qu'un processeur d'image.

Grâce au bus SCCB, il peut produire des données d'image en 8 ou 10 bits à plusieurs résolutions, que ce soit en plein format, par sous-échantillonnage, en zoom ou en fenêtrage. La fréquence d'images atteint 15 images par seconde en mode UXGA, tandis qu'elle peut aller jusqu'à 30 images par seconde pour le SVGA et 60 images par seconde pour le CIF. L'utilisateur a un contrôle complet sur la qualité de l'image, le format des données et leur transmission.

Toutes les fonctions de traitement d'image, comme la courbe gamma, la balance des blancs, le contraste et la chrominance, peuvent être programmées via l'interface SCCB. Ce capteur adopte une technologie spécifique pour améliorer la qualité visuelle, en réduisant ou en supprimant certains défauts optiques ou électroniques, tels que le bruit de motif fixe, le flou ou les images fantômes, ce qui permet d'obtenir des couleurs nettes et stables **(**figure 2.5**) [26].**



Figure.2.5. Camera OV2640 grand angle **[27].**

2.2.3. Ecran LCD (16X2) avec interface I2C :

2.2.3.1. Ecran LCD (16X2) :

LCD est l'abréviation de l'expression anglaise (**L**iquid **C**rystal **D**isplay), que l'on traduit en français par écran à cristaux liquides. C'est pourquoi on parle d'afficheur LCD. Cet afficheur sert surtout d'interface visuelle entre un système (ou un projet) et l'utilisateur. Sa fonction principale est de transmettre des informations importantes du système à l'utilisateur. Il affiche donc des données que l'utilisateur peut consulter et utiliser.

Le nombre de caractères affichables par ligne est de 16 pour ce modèle, répartis sur deux lignes, ce qui donne un total de 32 caractères. On trouve cette information dans la référence 1602 A.

On obtient aussi des détails sur le mode de transmission des données, qui peut se faire sur quatre (4) ou huit (8) bits :

- En mode 8 bits, les octets passent directement sur les lignes DB0 à DB7.
- En mode 4 bits, par contre, les octets sont envoyés en deux étapes, uniquement sur les lignes DB4 à DB7 **(**figure 2.6) **[28].**

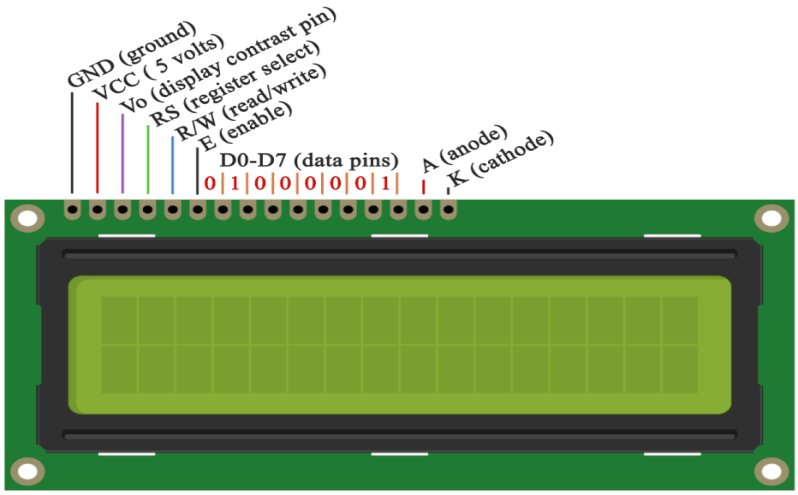


Figure.2.6. Ecran LCD **[29].**

2.2.3.2. Interface I2C:

Le bus I2C, ou Inter-Integrated Circuit, est un protocole de communication série qui permet à plusieurs puces de parler entre elles en utilisant seulement deux fils : une ligne de données **(SDA)** et une ligne d'horloge **(SCL)**, plus **(VCC)** et **(GND)**.

Cette approche simple est adaptée pour relier des circuits intégrés sur une même carte, surtout dans les systèmes embarqués où l'espace est limité. Le protocole s'appuie sur une architecture maître-esclave ; le maître gère la ligne d'horloge et lance la communication avec les esclaves. Chaque appareil sur le bus a une adresse unique, ce qui aide le maître à dialoguer avec chacun de manière ciblée. Ce système est reconnu pour être efficace et facile à mettre en place, ce qui explique son usage courant, de l'électronique grand public à l'industrie. Sa capacité à gérer plusieurs appareils avec un câblage réduit séduit les concepteurs qui cherchent à simplifier leurs projets tout en assurant une communication fiable entre composants **(**figure 2.7**) [30].**

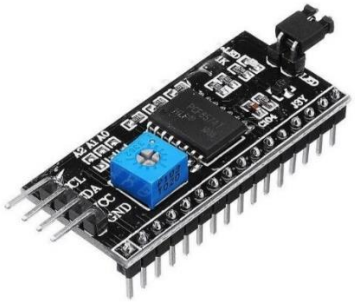


Figure.2.7. Interface I2C **[31].**

Cette section mentionne certaines des caractéristiques et spécifications du module adaptateur d'interface série I2C (figure 2.8) :

- Tension de fonctionnement : 5 V CC
- Contrôle I2C via PCF8574
- Il est possible d'avoir 8 modules sur un seul bus I2C.
- Adresse I2C : 0X20~0X27 (l'adresse d'origine est 0X20, vous pouvez la modifier vous-même via les cavaliers intégrés).

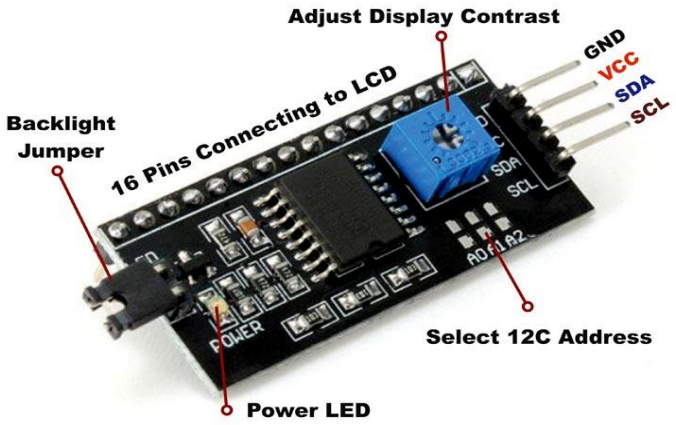


Figure.2.8. Caractéristiques du I2C **[32]**.

2.2.4. Servomoteur :

Un servomoteur est un appareil électrique qui sert à faire tourner des objets, comme un bras robotisé, avec une grande précision.

Il se compose d'un moteur à courant continu équipé d'un système de rétroaction négative qui permet de détecter les erreurs. Cela assure un contrôle précis de la vitesse de rotation et de la position du moteur. Parfois, on utilise aussi des moteurs à courant alternatif pour cette fonction.

Ce système fonctionne en boucle fermée, utilisant la rétroaction négative pour gérer le mouvement et la position finale de l'arbre. Il n'est pas conçu pour tourner en continu comme les moteurs AC/DC classiques. L'angle de rotation est généralement limité entre 0° et 180° (figure 2.9**) [33].**



Figure.2.9. Servo Moteur **[33].**

Le servomoteur possède trois broches pour son fonctionnement, comme le montre la figure ci-dessus (figure 2.10) :

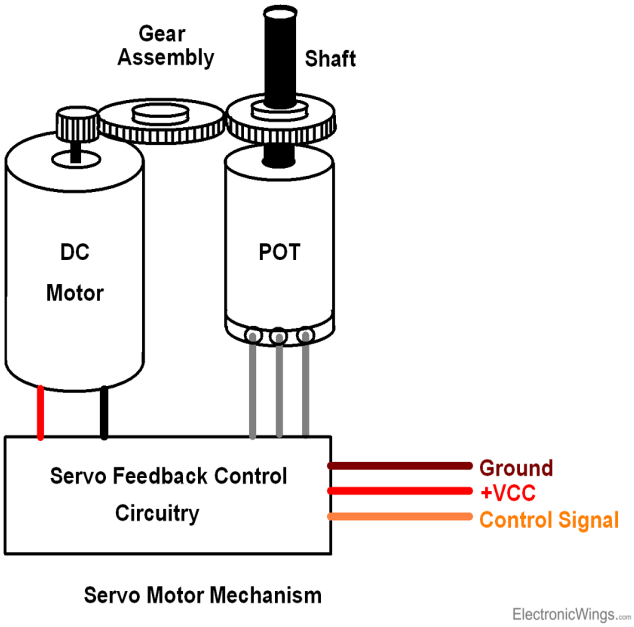


Figure.2.10. Mécanismes Du Servo Moteur **[33].**

- **+VCC :** Connectez l'alimentation +VCC à cette broche. Pour le micro-servo SG90, la tension est de 4,8 V (~5 V).

- **Sol :** Connectez la masse à cette broche.

- **Signal de contrôle :** Connectez un PWM de période 20 ms (50 Hz) à cette broche **[33]**.

2.2.5. Clavier Matriciels :

Le Clavier Matricé (**4×4)**, avec ses 16 touches, est un dispositif d'entrée compact et pratique. Il convient bien aux projets qui demandent une interface utilisateur pour saisir des données, qu'elles soient numériques ou alphanumériques. Ce clavier est structuré en une matrice de 4 lignes et 4 colonnes, ce qui facilite la détection des pressions sur les touches. Lorsqu'une touche est actionnée, elle établit un contact entre une ligne et une colonne, ce qui permet au microcontrôleur d'identifier clairement la touche enfoncée.

Le clavier matricé 4×4 fonctionne en balayant successivement ses lignes et ses colonnes pour repérer la touche appuyée. Quand une touche est pressée, elle connecte une ligne à une colonne précise, ce qui permet au microcontrôleur d'identifier cette touche. Ce balayage est souvent pris en charge par une bibliothèque logicielle dans des environnements comme Arduino, ce qui rend son utilisation plus simple dans vos projets.

Spécifications Techniques :

- Configuration : 4 lignes x 4 colonnes.
- Nombre de touches : 16.
- Type de contact : Membrane, offrant une action tactile fiable et durable.
- Tension de fonctionnement : 5V DC, compatible avec la majorité des systèmes électroniques, y compris Arduino.
- Dimensions : 70mm x 77mm, un format compact pour une intégration facile dans différents projets.
- Durée de vie : Jusqu'à 1 million de pressions par touche, garantissant une grande fiabilité et une utilisation à long terme.
- Interface : Le clavier est équipé de 8 broches (4 pour les lignes, 4 pour les colonnes) pour une connexion simple aux microcontrôleurs et aux plateformes de prototypage (figure 2.11) **[34]**.

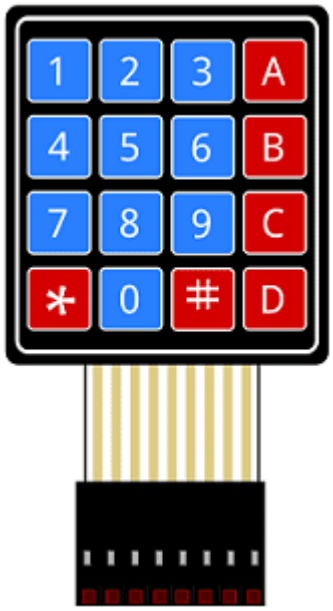


Figure.2.11. Clavier Matriciels **[35]**.

2.2.6. Code QR :

Un code QR, en anglais (**Q**uick **R**esponse) ou code à réponse rapide, est un type de code-barres pensé pour être facilement lu par les applications et les caméras des smartphones. Ce qui rend les codes QR intéressants, c'est leur utilisation variée : ils peuvent contenir des coordonnées, offrir des coupons de réduction, rediriger vers des sites web, inviter à laisser un avis, connecter à des événements Facebook, faciliter des achats, entre autres choses **[36]**.

Un QR code est un motif carré composé de points et de lignes en noir et blanc, dans lequel des informations sont codées, (figure.2.13 et 2.14). Ce carré peut contenir jusqu'à 177 sur 177 éléments, ce qui correspond environ à une demi-page de texte au format A4. En termes de capacité, un QR code peut stocker jusqu'à 4 296 caractères alphanumériques ou 7 089 chiffres décimaux si on utilise le niveau de correction d'erreur le plus faible. Cependant, plus ce niveau de correction est élevé, plus la capacité du QR code diminue.

Un QR code se compose de deux éléments (figure.2.13) :

- Trois carrés aux coins du QR code, destinés au scannage
- Le QR code lui-même, qui est intégré comme un modèle et contient les informations **[37]**.



Figure.2.12. Code QR **[38]**.

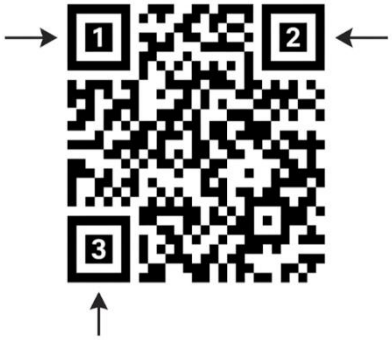


Figure.2.13. Les composants du QR.

2.3. Logiciels et applications utilisés :

2.3.1. Arduino IDE :

Le logiciel Arduino est un environnement de développement intégré (IDE) ouvert et gratuit, que l'on peut télécharger directement depuis le site officiel d'Arduino.

Ce logiciel permet d'écrire un programme, appelé croquis (ou sketch en anglais). Ces programmes sont rédigés en langage C.

Il offre aussi la possibilité de compiler ce programme pour le traduire en langage machine, compréhensible par le microcontrôleur de l'Arduino. La zone en bas de la fenêtre affiche les informations sur cette compilation, ainsi que les éventuels messages d'erreur ou notifications.

Le programme peut ensuite être téléversé dans la mémoire de l'Arduino via le port USB de l'ordinateur. Une fois chargé, on parle de microgiciel. Là encore, la zone en bas informe sur le déroulement du téléversement et montre les erreurs ou autres messages utile

Il est possible de communiquer avec la carte Arduino grâce au terminal, aussi appelé moniteur série. Pendant que le programme fonctionne dans la mémoire de l'Arduino, la carte peut échanger des données avec l'ordinateur tant que la connexion USB est active **[39]**.

La figure ci dessue (figure2.15) montre interface de Arduino IDE :

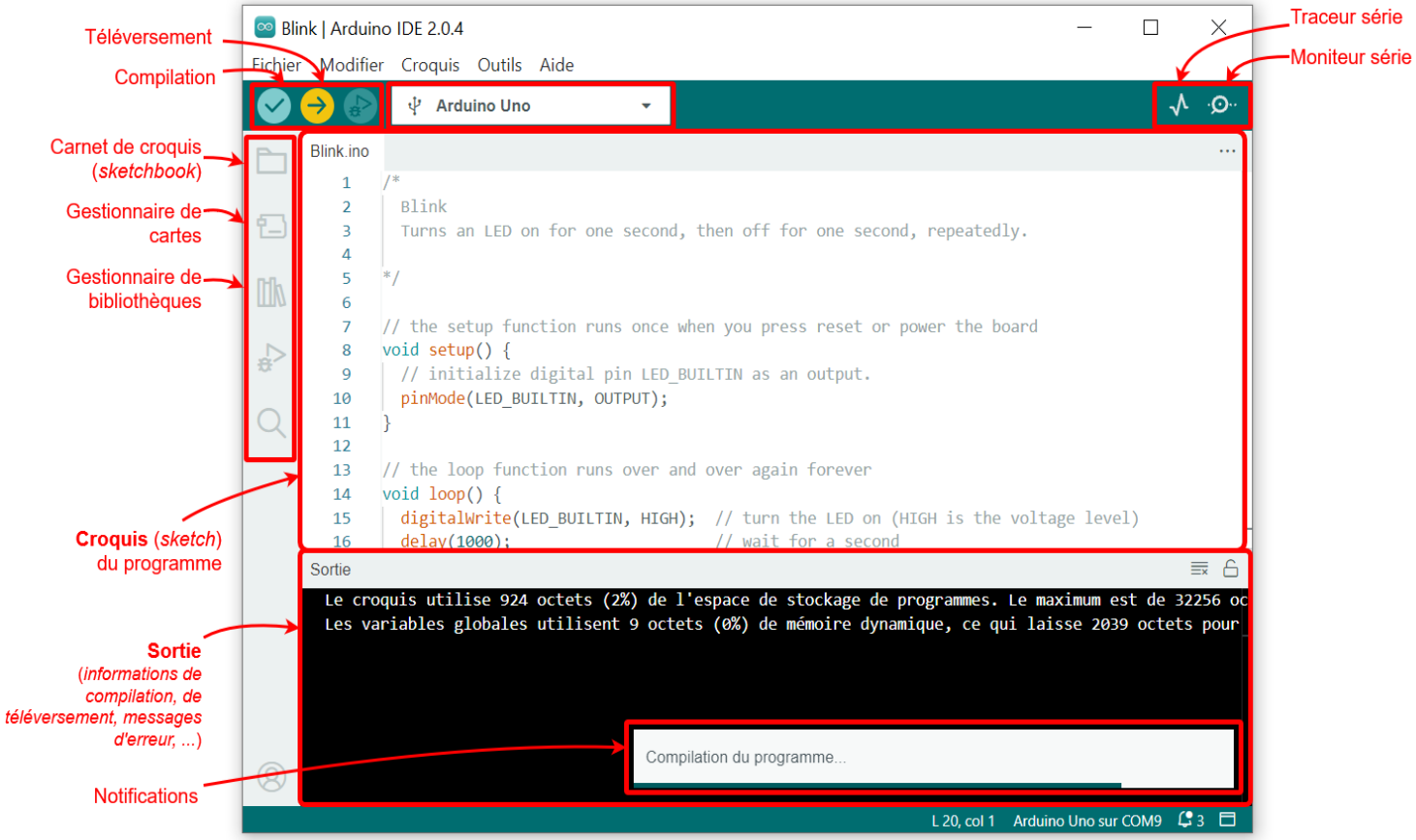


Figure.2.14. Interface du Arduino IDE **[39]**.

- **La Zone d'Édition (Croquis / Sketch) :**

C'est l'espace principal où vous écrivez votre code source. Un programme Arduino, appelé croquis, est rédigé en C++. Il comprend généralement deux fonctions indispensables : setup (), qui sert à initialiser les composants au démarrage, et loop (), qui contient les instructions répétées tant que la carte reste alimentée.

- **Bouton de Compilation (Vérifier) :**

Représenté par une coche, ce bouton sert à vérifier la syntaxe de votre code sans l'envoyer à la carte. Le logiciel traduit votre code en langage machine et affiche des messages d'erreur en bas de l'écran s'il détecte des erreurs comme un point-virgule oublié ou une variable mal déclarée.

- **Bouton de Téléversement :**

Sous la forme d'une flèche pointant vers la droite, ce bouton lance d'abord une compilation puis transfère le programme dans la mémoire flash du microcontrôleur via le câble USB. Une fois terminé, la carte exécute automatiquement votre code.

- **Le Carnet de Croquis (Sketchbook) :**

Cette icône en forme de dossier donne accès à votre bibliothèque perso de projets. C'est là que sont rangés tous vos fichiers. ino, ce qui facilite le passage d'un projet à l'autre ou la réutilisation de morceaux de code venant d'anciens travaux.

- **Le Gestionnaire de Cartes :**

Ce menu est essentiel pour configurer l'IDE en fonction du matériel utilisé. Par exemple, si vous avez un Arduino Mega ou une Uno, il faut choisir le modèle précis dans ce gestionnaire pour que le logiciel puisse bien communiquer avec le processeur de votre carte.

- **Le Gestionnaire de Bibliothèques :**

Les bibliothèques regroupent des codes pré-écrits par la communauté. Ce panneau vous permet de chercher et d'installer des extensions pour piloter facilement des composants complexes, comme un capteur ultrason, un module LiFi ou un écran LCD, sans devoir coder toutes leurs fonctions de base.

- **Le Moniteur Série :**

C'est une fenêtre de dialogue en texte. Elle permet à la carte Arduino d'envoyer des informations à votre ordinateur (par exemple, afficher la valeur d'un capteur) ou de recevoir des commandes depuis votre clavier. C'est l'outil principal pour déboguer et vérifier que votre code fonctionne.

- **Le Traceur Série :**

À la différence du moniteur qui affiche du texte, le traceur série transforme les données numériques en graphiques en temps réel. Il est très utile pour visualiser l'évolution d'un signal analogique, comme les variations de lumière captées par un capteur LDR dans un projet de communication optique.

- **La Console de Sortie :**

Située en bas, cette zone noire montre les détails techniques de l'opération en cours. Elle indique la mémoire utilisée par votre programme et, en cas d'échec, donne des informations précises sur l'origine du problème (erreur de port COM, erreur de compilation, etc.).

- **Barre d'État Inférieure**

Cette petite bande en bas de la fenêtre affiche en un coup d'œil deux infos importantes : le modèle de la carte connectée et le numéro du port série (COM) utilisé. Si ces infos ne correspondent pas à votre connexion réelle, le téléversement ne fonctionnera pas.

2.3.2. Application MacroDroid :

MacroDroid est une application android qui se trouve sur **Google Play Stor,** permet d'automatiser des actions sur smartphone, ces actions sont appelées des Macros. Dans les macros, on inclut des informations regroupées en trois catégories principales : les déclencheurs, les actions et les contraintes. Les déclencheurs correspondent à des événements repérés par le téléphone qui vont démarrer la macro. Les actions désignent ce que la macro effectue. Quant aux contraintes, elles servent à limiter l'exécution de la macro aux seules situations où ces conditions sont remplies **[40]**.

Par exemple je veux allumer lampe du porche ou faire une appelle où afficher une photo, il suffit d'accéder aux paramètres de l'application et de sélectionner la technologie à mettre en œuvre. Nous programmons ensuite l'application pour qu'elle effectue l'action souhaitée, avec certaines limitations, et ce, d'un simple clic sur une petite icône à l'écran du téléphone. Voir la figure ci dessue (figure.2.16) :

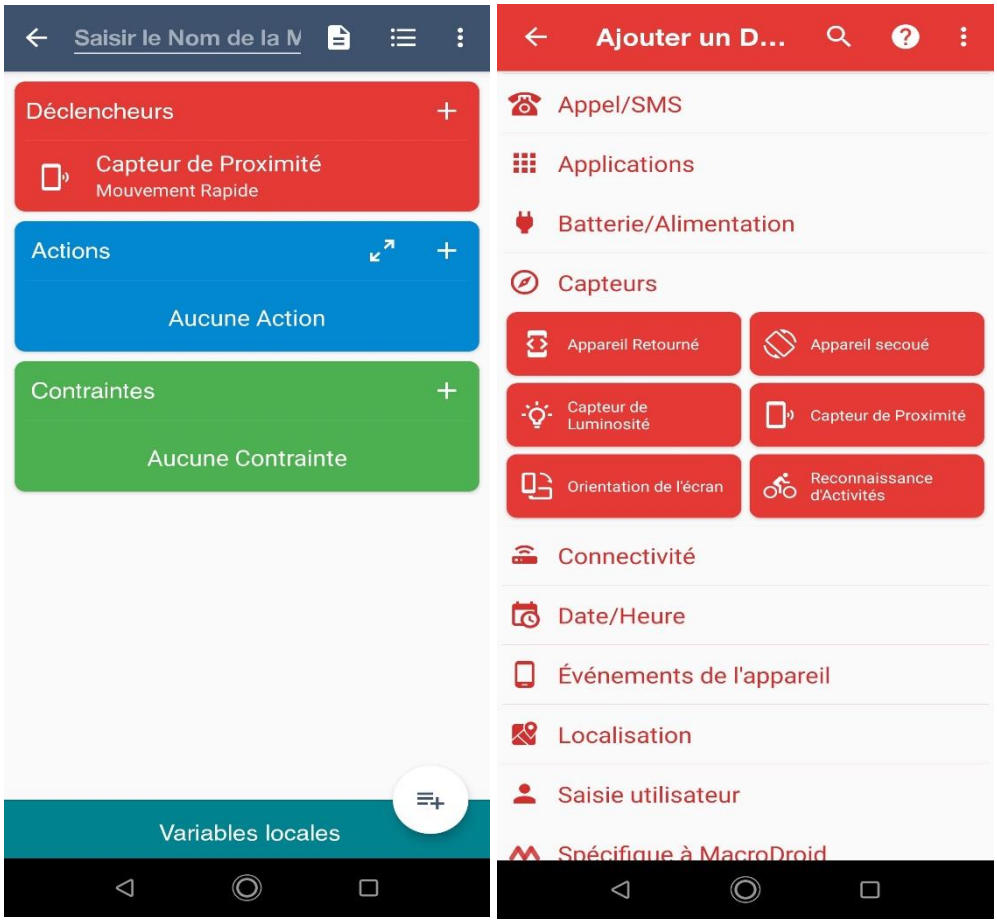


Figure.2.15. Interface de paramètres de MacroDdroid **[40]**.

Section Modèle s'avère aussi très pratique. On y trouve des macros partagées par tous les utilisateurs disposant d'une version non limitée. Vous pourrez y dénicher de nombreux exemples intéressants qui vous aideront à concevoir vos propres macros.

Voilà un aperçu rapide de cet outil remarquable. Ce qui reste le plus stimulant, c'est de réfléchir intensément pour identifier des solutions adaptées à votre problème **[40]**.

2.4. Conclusion :

Dans ce chapitre, nous avons décrit les outils et logiciels utilisés pour la partie pratique, en insistant plus particulièrement sur Arduino IDE et le code QR, qui jouent un rôle central dans le projet. Le chapitre suivant expliquera plus en détail la mise en œuvre pratique et exposera les résultats obtenus.

Chapitre 3

Réalisation et mise en œuvre d'un système d'accès et de contrôle

3.1. Introduction :

Ce chapitre est consacré à la présentation détaillée du système d'identification et de contrôle d'accès sécurisé, tel qu'il a été élaboré pour ce projet. Nous y aborderons les différentes phases de sa conception, puis les étapes de sa mise en œuvre. Pour cela, nous prendrons comme référence les composants matériels et logiciels que nous avons décrits dans le chapitre précédent.

3.2. Description du projet :

Ce projet développe un système de contrôle d'accès sécurisé destiné aux résidences, fondé sur l'utilisation de codes QR pour l'identification plus un code PIN pour double la sécurité. L'infrastructure est conçue autour de dispositifs qui allient économie, performance et compacité. L'objectif principal de cette initiative est d'assurer un niveau adéquat de sécurité et de contrôle, tant pour les environnements domestiques que professionnels.

3.3. Conception et réalisation :

Initialement, nous avons conçu une configuration matérielle intégrant l'ensemble des dispositifs précédemment cités. Ces éléments ont ensuite été programmés à travers l'environnement de développement Arduino IDE, avec pour objectif de réguler le fonctionnement du système, d'assurer les mesures de sécurité nécessaires et de gérer l'ouverture des accès.

3.3.1. Programmation les microcontrôleurs :

Premièrement, nous programmons les unités de contrôle afin de garantir la séquence et la précision du système. Pour cela, nous téléchargeons les bibliothèques nécessaires depuis le site **GitHub.com** vers le **disque local (c :),** puis nous les intégrons dans l'environnement Arduino IDE (voir figure 1.3). Cette étape nous permet de programmer les différentes unités de contrôle. Parmi ces bibliothèques, on peut citer :

- **Quirc.h**: bibliothèque utilisée comme moteur de décodage des QR codes. Elle analyse la matrice de pixels capturée par la caméra afin de détecter les carrés de position du QR code et de convertir les données en texte.
- **LiquidCrystal_I2C.h** : Permet de contrôler l'affichage du texte sur ton écran 16x2.
- **Keypad.h** : Gère la matrice de touches pour détecter quel chiffre ou quelle lettre est pressé.

- **Servo.h** : Bibliothèque standard pour contrôler l'angle du servomoteur (de **0°** à **180°**).

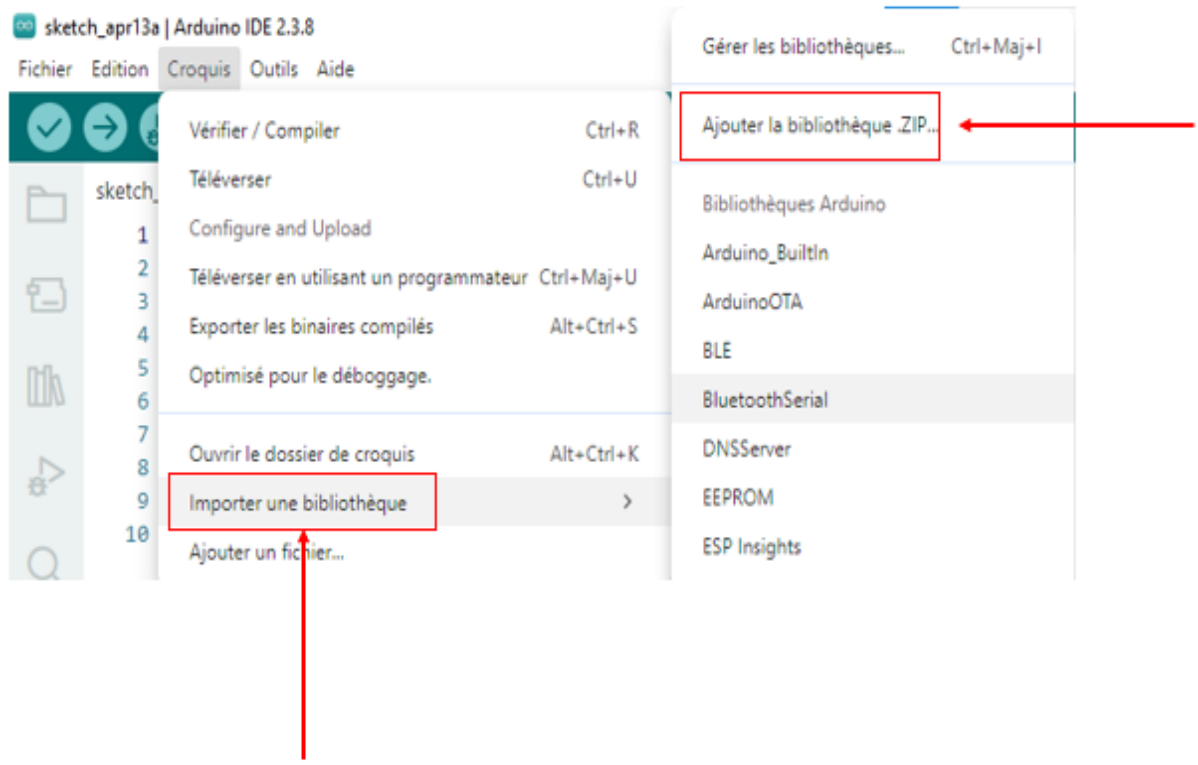


Figure.3.1. Les étapes pour ajouter les bibliothèques sur Arduino IDE.

Après l'installation des bibliothèques requises, nous procédons à la rédaction des codes du programme destiné aux unités de contrôle, en vue de leur configuration pour le système d'accès. Nous commençons par :

3.3.2.1. ESP32 :

À l'aide du logiciel Arduino IDE et en utilisant la bibliothèque **quirc.h**, nous avons programmé l'unité de contrôle ESP32 avec un programme spécial capable de capturer les codes QR, d'analyser leurs données, de les afficher, puis de les transférer directement à l'unité de contrôle Arduino Mega en un court laps de temps.

D'abord, dans l'interface du logiciel, nous sélectionnons en haut le type de carte utilisé — **ESP32 Wrover Kit** (figure 3.2) —, puis nous saisissons le code du programme pour l'analyse et la lecture des codes QR (figures 3.3 à 3.7).



Figure.3.2. Sélection du type de carte utilisée (ESP32 Wrover kit).

Après avoir sélectionné le type de carte à utiliser, nous écrivons un code qui permet à l'ESP32-CAM de capturer une image, d'y détecter un code QR et de le décoder pour en extraire les données. Il compare ensuite le contenu lu à une valeur prédéfinie afin d'envoyer un signal d'autorisation ou de refus à la carte Arduino Mega pour l'ouverture de la porte.

Ce code est structuré autour de quatre fonctions principales nécessaires au fonctionnement du système de vision (figure 3.3 …. 3.7) :

1. **Définition du brochage (Pins) :** Il établit la connexion physique entre le microcontrôleur ESP32 et les broches du capteur caméra OV2640.

2. **Configuration matérielle :** Il initialise le capteur en réglant la résolution de l'image (format QVGA) et la qualité du flux pour optimiser la reconnaissance.

3. **Algorithme de traitement :** Il utilise la bibliothèque de reconnaissance optique pour analyser l'image capturée, localiser le QR code et extraire le texte qu'il contient.

4. **Logique de décision :** Il vérifie si le texte décodé correspond à l'identifiant autorisé (PFE2026) et transmet le résultat à l'Arduino via une liaison série.

- **Inclusion des bibliothèques** : Cette partie importe les outils nécessaires pour que l'ESP32 puisse utiliser sa caméra intégrée et comprendre le langage des QR codes. Elle inclut notamment la bibliothèque (figure 3.3).

ESP32 Wrover Kit (all versi...

qrcodereader.ino

```
1   #include "esp_camera.h"
2   #include "soc/soc.h"
3   #include "soc/rtc_cntl_reg.h"
4   #include "quirc.h"
5
6   /* ===================================== CONFIGURATION ESP32-CAM AI-THINKER */
7   TaskHandle_t QRCodeReader_Task;
8
9     #define PWDN_GPIO_NUM    -1
10    #define RESET_GPIO_NUM   -1
11    #define XCLK_GPIO_NUM    21
12    #define SIOD_GPIO_NUM    26
13    #define SIOC_GPIO_NUM    27
14
15    #define Y9_GPIO_NUM      35
16    #define Y8_GPIO_NUM      34
17    #define Y7_GPIO_NUM      39
18    #define Y6_GPIO_NUM      36
19    #define Y5_GPIO_NUM      19
20    #define Y4_GPIO_NUM      18
21    #define Y3_GPIO_NUM       5
22    #define Y2_GPIO_NUM       4
23    #define VSYNC_GPIO_NUM   25
24    #define HREF_GPIO_NUM    23
25    #define PCLK_GPIO_NUM    22
26
```

Figure.3.3. Code programme du ESP32.

- **Configuration des broches / Pins** :Comme l'ESP32-CAM possède de nombreuses connexions internes, ce bloc de code définit précisément quelles broches sont reliées au capteur d'image (OV2640). C'est le "plan de câblage" logiciel du capteur (figure 3.5).

ESP32 Wrover Kit (all versi...

qrcodereader.ino

```
/* ===================================== VARIABLES */
struct quirc *q = NULL;
camera_fb_t * fb = NULL;
struct quirc_code code;
struct quirc_data data;

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);
  Serial.println("\n--- SCANNER QR (CAMERA_MODEL_AI_THINKER) ---");

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
```

Figure.3.4. Code programme du ESP32.

- **Réglages de l'image :** Ce segment configure la qualité de la photo. Il définit la résolution (généralement QVGA pour être rapide) et ajuste la luminosité ou le contraste pour que le QR code soit le plus lisible possible par l'algorithme (figure 3.5).



Figure.3.5. Code programme du ESP32.

- **Initialisation et Démarrage** : Dans le void setup(), le code vérifie si la caméra est bien présente et fonctionnelle. Si tout est correct, il lance le système de capture ; sinon, il affiche un message d'erreur sur le moniteur série (figure 3.6).

ESP32 Wrover Kit (all versi...

qrcodereader.ino

```
83   xTaskCreatePinnedToCore(QRCodeReader, "QR_Task", 15000, NULL, 1, &QRCodeReader_Task, 0);
84
85   Serial.println("Prêt ! Présentez le QR Code.");
86  }
87
88  void loop() { delay(1); }
89
90  void QRCodeReader( void * pvParameters ){
91    while(1){
92      fb = esp_camera_fb_get();
93      if (!fb) { vTaskDelay(10); continue; }
94
95      uint8_t *image_buffer = quirc_begin(q, NULL, NULL);
96
97      // --- FILTRE BINAIRE POUR 120° ---
98      uint8_t minV = 255, maxV = 0;
99      for (int i = 0; i < fb->len; i++) {
100       if (fb->buf[i] < minV) minV = fb->buf[i];
101       if (fb->buf[i] > maxV) maxV = fb->buf[i];
102     }
103     uint8_t threshold = minV + (maxV - minV) / 2;
104
105     for (int i = 0; i < fb->len; i++) {
106       image_buffer[i] = (fb->buf[i] < threshold) ? 0 : 255;
107     }
108
109     quirc_end(q);
110
```

Figure.3.6. Code programme du ESP32.

• **Décodage et Transmission :** C'est la boucle principale. La caméra prend une image, l'analyse pour trouver un QR code, et si elle en détecte un, elle extrait le texte. Elle envoie ensuite un signal (le succès) vers l'Arduino Mega pour lui dire d'ouvrir la porte (figure 3.7).

```
111     int count = quirc_count(q);
112     if (count > 0) {
113       quirc_extract(q, 0, &code);
114       if (quirc_decode(&code, &data) == QUIRC_SUCCESS) {
115         Serial.println("\n QR CODE VALIDE !");
116         Serial.print((const char *)data.payload);
117         vTaskDelay(2000 / portTICK_PERIOD_MS);
118       } else {
119         Serial.println("Échec lecture (Erreur optique)");
120       }
121     }
122
123     esp_camera_fb_return(fb);
124     vTaskDelay(30 / portTICK_PERIOD_MS);
125   }
126 }
```

Figure.3.7. Code programme du ESP32.

Une fois le programme écrit, un message apparaît indiquant sur le moniteur série que ESP32 est prête à capturer et à analyser le code QR (figure 3.8).

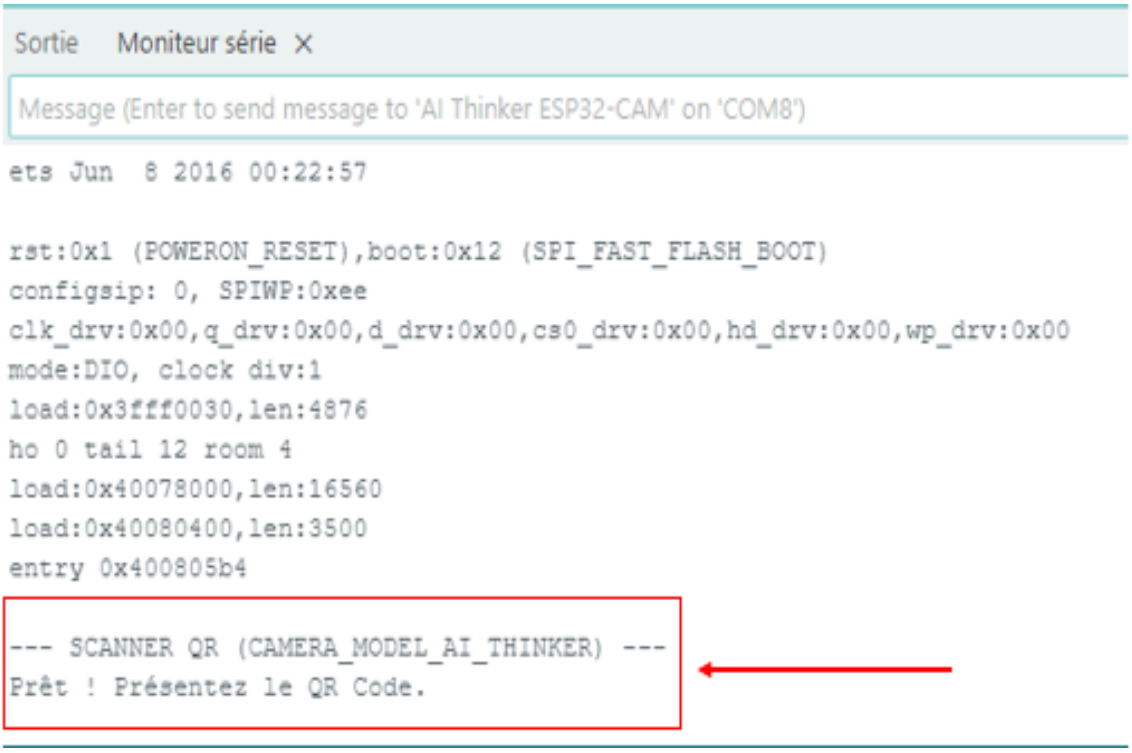


Figure.3.8. Affichage du message que ESP32 est prêt.

3.3.2.2. Arduini Mega 2560 :

Une fois la programmation de l'ESP32 terminée, nous passons à la programmation de l'Arduino Mega 2560. En suivant les mêmes étapes qu'auparavant, nous sélectionnons le type de carte (figure 3.10), puis nous saisissons le code complet du programme (figures 3.11 à 3.15), lequel permet de contrôler l'ouverture et la fermeture de la porte, d'allumer la LED et d'afficher le texte sur l'écran LCD.

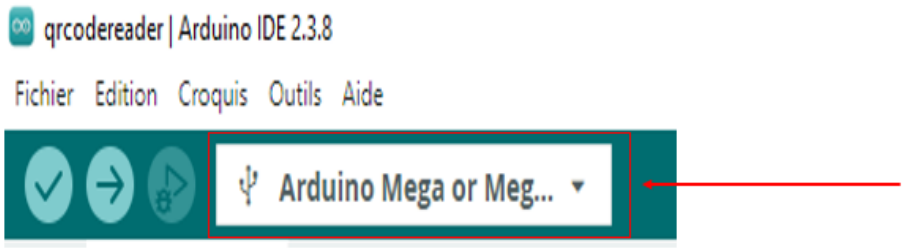


Figure 3.9. Sélection du type de carte utilisée (Arduino Mega).

- **Bibliothèques :** Cette partie sert à importer les "outils" nécessaires. On appelle les bibliothèques qui permettent à l'Arduino de parler avec l'écran LCD, de lire les touches du clavier matriciel et de faire tourner le servomoteur de la porte (figure 3.10).

ouvrirlaporte.ino

```
#include <Keypad.h>
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// --- CONFIGURATION MATÉRIEL ---
LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo myServo;

const int pinRouge = 2;   // LED Rouge (Accès refusé / Attente)
const int pinVert = 3;    // LED Verte (Accès accordé)
const int pinServo = 10;  // Signal du Servomoteur

// --- LOGIQUE DE SÉCURITÉ ---
String qrAutorise = "PFE2026"; // Le contenu du QR code attendu
char pinAutorise[] = "1234";    // Le code PIN attendu
char pinSaisi[5];               // Stocke les 4 chiffres tapés + caractère nul
int indexPin = 0;
bool qrEstValide = false;       // Flag de progression

// --- CONFIGURATION CLAVIER (Pins 22-29) ---
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
```

Figure.3.10. Code de programmation du Arduino Mega.

- **Configuration matérielle :** Dans cette section, nous définissons le plan de branchement. Nous indiquons à l'Arduino exactement sur quelles broches sont connectés

l'écran et le clavier, et nous définissons la cartographie du clavier (les chiffres de 0 à 9 et les lettres A à D) afin que le programme reconnaisse chaque touche pressée (figure 3.11).

```
byte rowPins[ROWS] = {22, 23, 24, 25};
byte colPins[COLS] = {26, 27, 28, 29};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  // Initialisation des communications
  Serial.begin(115200);   // Vers le PC pour le débug
  Serial1.begin(115200);  // Vers l'ESP32-CAM (Pin 19 RX1)

  // Configuration des sorties
  pinMode(pinRouge, OUTPUT);
  pinMode(pinVert, OUTPUT);
  myServo.write(0);        // <--- FORCE LA POSITION 0 AVANT
  delay(50);               // Laisse le temps au signal de se stabiliser
  myServo.attach(pinServo);

  // Initialisation LCD
  lcd.init();
  lcd.backlight();

  resetSystem();
  Serial.println("Système MEGA prêt. En attente de l'ESP32...");
}

void loop() {
  // --- PARTIE 1 : RÉCEPTION DU QR CODE (ESP32 -> MEGA) ---
  if (!qrEstValide && Serial1.available() > 0) {
```

Figure.3.11. Code de programmation du Arduino Mega.

- **Initialisation / Setup :** C'est la phase de préparation qui s'exécute une seule fois au démarrage. On active la communication avec l'ordinateur et l'ESP32, on prépare l'écran LCD pour l'affichage, et on s'assure que la porte est bien fermée (servomoteur à 0°) (figure 3.12).

ouvrirlaporte.ino

```
58      String message = Serial1.readStringUntil('\n');
59      message.trim(); // Enlever les espaces et \r
60
61      if (message.length() > 0) {
62        Serial.print("QR Scanné : ");
63        Serial.println(message);
64
65        if (message == qrAutorise) {
66          qrEstValide = true;
67          lcd.clear();
68          lcd.print("QR CORRECT !");
69          digitalWrite(pinRouge, LOW); // On éteint le rouge pour montrer la progression
70          delay(1500);
71          lcd.clear();
72          lcd.print("ENTRER PIN:");
73        } else {
74          lcd.clear();
75          lcd.print("QR INCORRECT");
76          alerteErreur();
77          resetSystem();
78        }
79      }
80    }
81
82    // --- PARTIE 2 : SAISIE DU CODE PIN (Si QR déjà validé) ---
83    if (qrEstValide) {
84      char key = keypad.getKey();
85      if (key) {
```

Figure.3.12. Code de programmation du Arduino Mega.

- **Communication avec l'ESP32** : Ce bloc s'occupe du dialogue entre les deux cartes. L'Arduino reste à l'écoute : dès qu'il reçoit un signal positif (le chiffre '1') envoyé par l'ESP32-CAM après un scan réussi, il autorise l'étape suivante sur l'écran (figure 3.13).

ouvrirlaporte.ino

```
86          if (key == '*') { // Touche d'annulation
87            resetSystem();
88            return;
89          }
90
91          if (indexPin < 4) {
92            pinSaisi[indexPin] = key;
93            lcd.setCursor(indexPin, 1);
94            lcd.print('*'); // Masquage du code
95            indexPin++;
96
97            if (indexPin == 4) {
98              delay(500);
99              verifierTout();
100           }
101         }
102       }
103     }
104   }
105
106   void verifierTout() {
107     pinSaisi[4] = '\0'; // Terminer la chaîne de caractères
108
109     if (strcmp(pinSaisi, pinAutorise) == 0) {
110       // --- ACCÈS TOTAL ACCORDÉ ---
111       lcd.clear();
112       lcd.print("PIN CORRECT");
113       lcd.setCursor(0, 1);
```

Figure.3.13. Code de programmation du Arduino Mega.

- **Logique du Code PIN** : C'est la sécurité finale. Le programme enregistre les chiffres tapés sur le clavier. Si le code correspond au mot de passe enregistré (ex : 1234), il déclenche l'ouverture physique de la porte en faisant pivoter le servomoteur (figure 3.14).

ouvrirlaporte.ino

```
123       // --- ERREUR PIN ---
124       lcd.clear();
125       lcd.print("PIN ERREUR !");
126       alerteErreur();
127       resetSystem();
128     }
129   }
130
131   void alerteErreur() {
132     for(int i=0; i<3; i++) {
133       digitalWrite(pinRouge, HIGH); delay(200);
134       digitalWrite(pinRouge, LOW);  delay(200);
135     }
136   }
137
138   void resetSystem() {
139     qrEstValide = false;
140     indexPin = 0;
141     memset(pinSaisi, 0, sizeof(pinSaisi));
142
143     myServo.write(0);         // Fermeture moteur
144     digitalWrite(pinRouge, HIGH); // État d'attente
145     digitalWrite(pinVert, LOW);
146
147     lcd.clear();
148     lcd.setCursor(0, 0);
149     lcd.print("SCANNER QR...");
150   }
```

Figure.3.14. Code de programmation du Arduino Mega.

Après avoir saisi le code du programme, nous cliquons sur Téléverser afin de programmer l'Arduino Mega. Un message s'affiche alors, indiquant que l'Arduino est prêt à recevoir les informations transmises par l'ESP32 (figure 3.15).

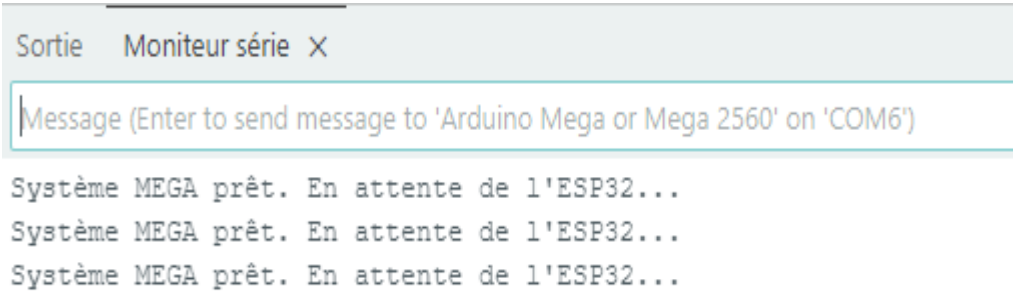


Figure.3.15. Affichage du moniteur de l'Arduino.

3.3.3. Création du code QR :

Une fois la programmation des unités de contrôle terminée, nous générons un code QR à l'aide du site web www.qr-code-generator.com, sur lequel nous saisissons la valeur prédéfinie utilisée lors de la programmation (PFE2026), puis nous le téléchargeons (figure 3.16).

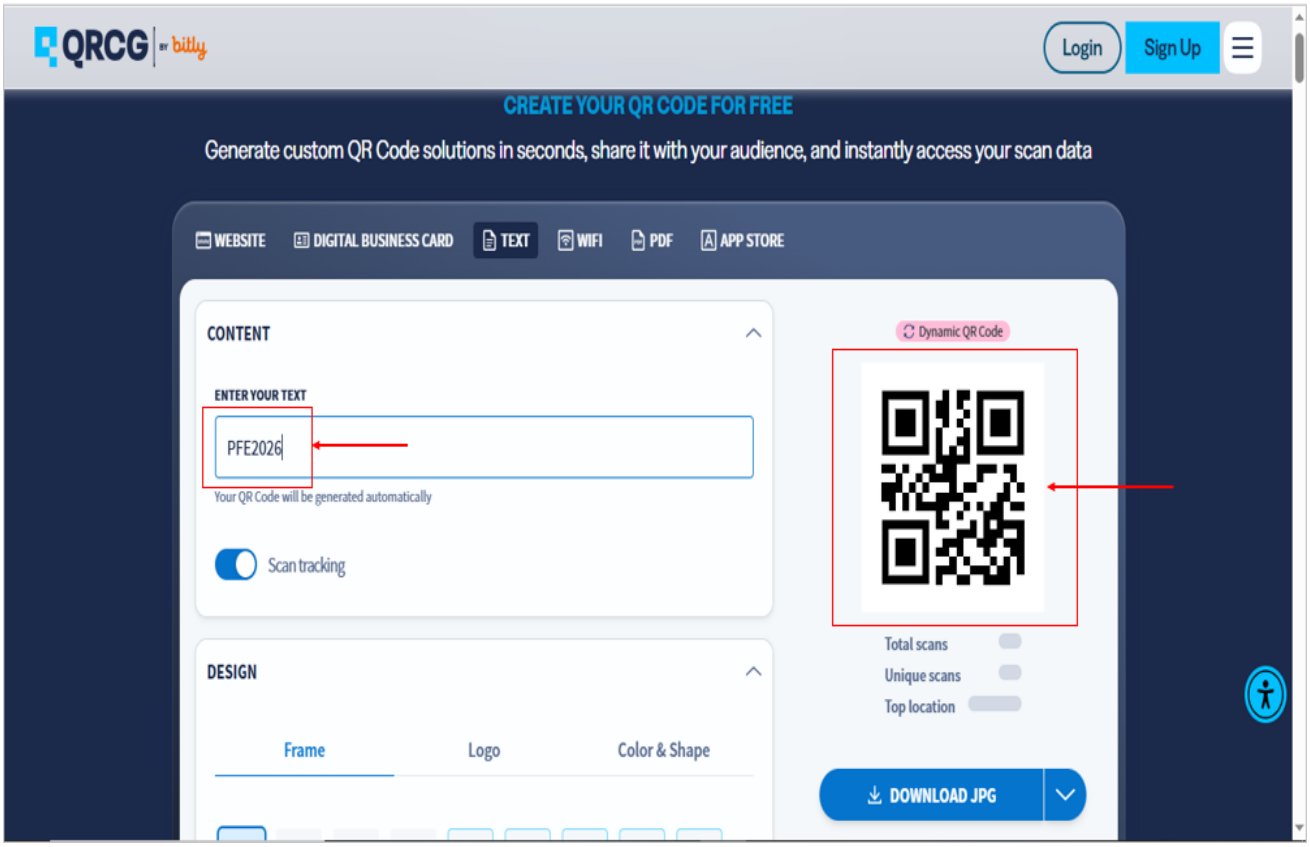


Figure.3.16. Création du code QR.

Après avoir généré le code QR, nous le téléchargeons sur le téléphone. Pour l'afficher rapidement devant la caméra, nous utilisons l'application **MacroDroid**, qui simplifie le processus d'affichage du code sur l'écran du téléphone mobile, en suivant les étapes ci-dessous (figures 3.17 à 3.26).

Nous ouvrons l'application installée sur le téléphone, puis dans l'interface, nous sélectionnons l'option **Ajouter une macro** et nous suivons les étapes suivantes :

- **Choix du mode d'action :** Nous sélectionnons la première option afin de préciser le type d'action à effectuer (figure 3.17).

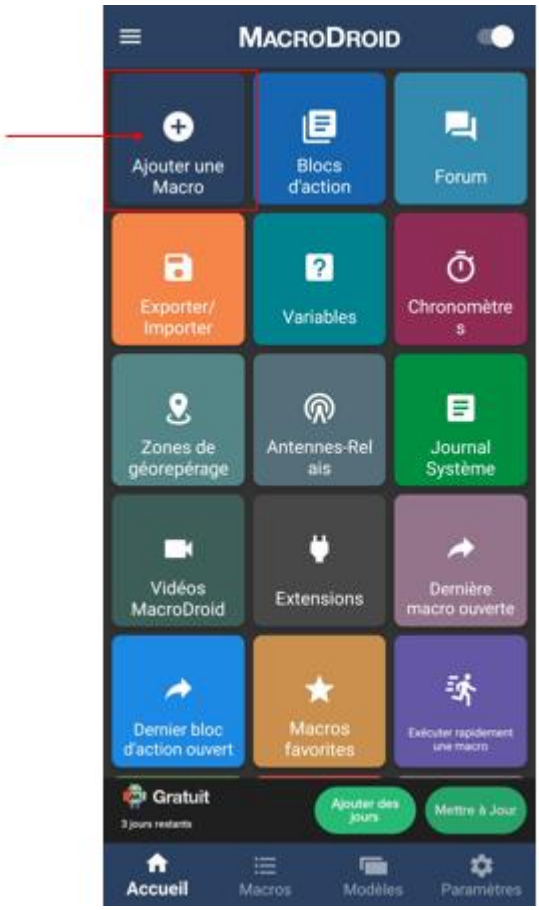


Figure 3.17. Accès à l'interface de création d'une nouvelle macro.

- **Initialisation de l'action :** Cette étape consiste à accéder au menu des **"Déclencheurs"**

Un déclencheur est l'événement qui va provoquer une action dans le système par exemple : "quand j'appuie sur ce bouton, la torche s'allume" (figure 3.18).



Figure 3.18. Sélection et configuration du déclencheur d'événement.

- **Choix du type de commande :** Dans cette section, nous définissons la manière dont l'utilisateur interagit avec l'application. En sélectionnant **Saisie utilisateur** puis **Bouton flottant**, nous créons un bouton virtuel qui restera visible au premier plan de l'écran du smartphone pour un accès rapide (figure 3.19).

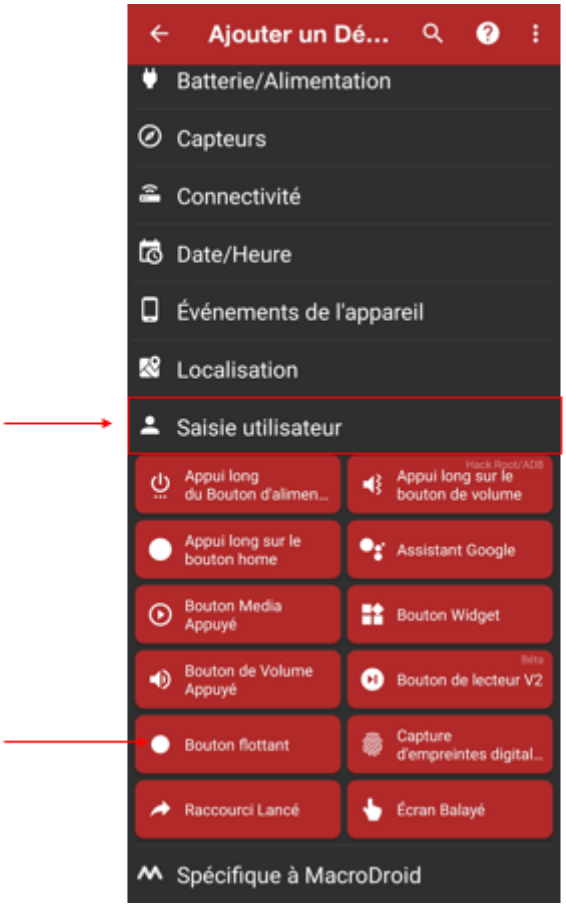


Figure 3.19. Définition du mode d'interaction via le bouton flottant.

- **Personnalisation visuelle :** Cette phase concerne le **design de l'interface**. On choisit une icône représentative (par exemple une icône de clé ou de porte) pour que l'utilisateur comprenne immédiatement la fonction du bouton avant de valider la création (figure 3.20).

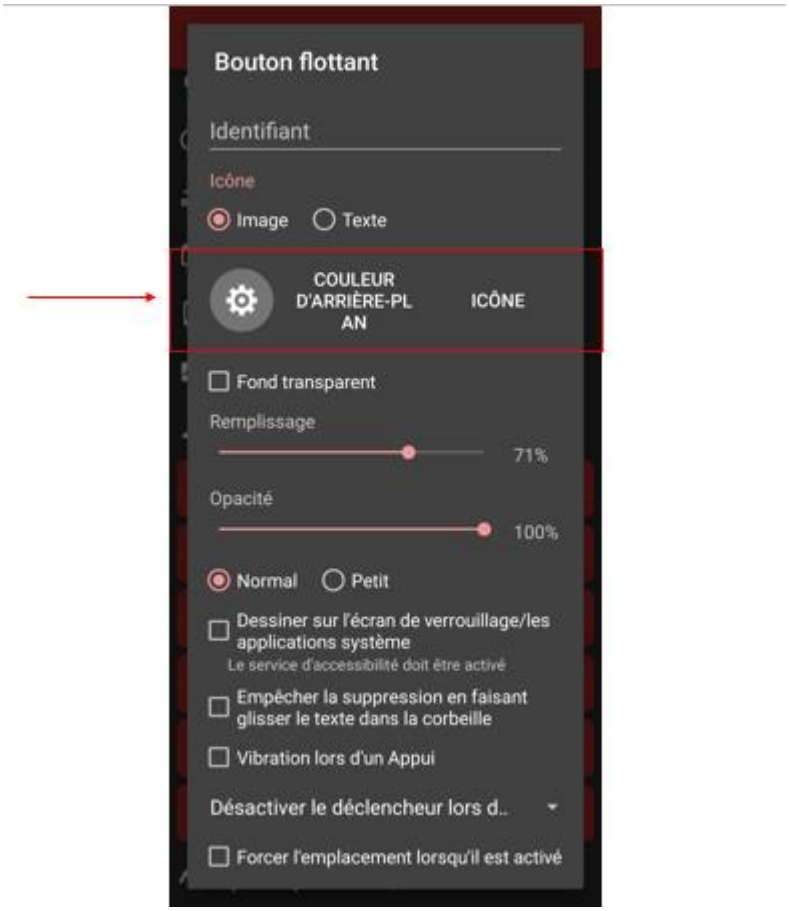


Figure 3.20. Personnalisation de l'icône de commande pour l'interface utilisateur.

- **Finalisation :** Il s'agit de l'étape de confirmation, qui enregistre la configuration et place le bouton de commande sur l'interface finale de l'utilisateur.

Ensuite, afin d'afficher le code QR précédemment téléchargé lors de l'appui sur le bouton, nous sélectionnons l'option **Action** (figure 3.21).

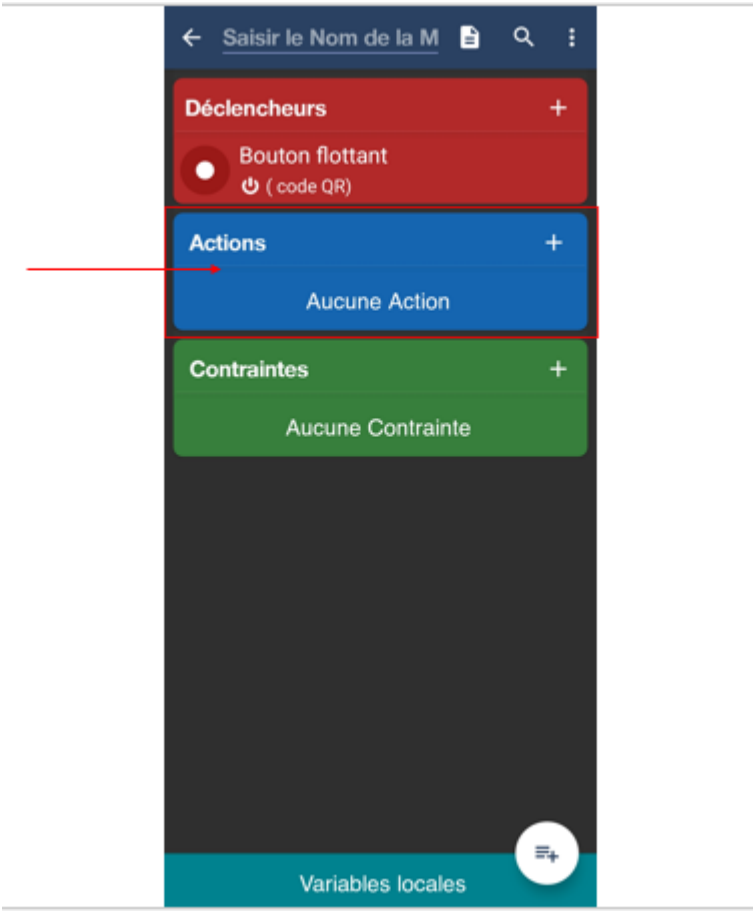


Figure 3.21. Sélection de la catégorie d'action à associer au bouton.

• **Accès au stockage :** Cette étape consiste à lier l'action du bouton à une ressource stockée sur le téléphone. En sélectionnant **Fichiers** puis **Ouvrir un fichier**, nous définissons que l'appui sur le bouton doit déclencher l'ouverture d'un document spécifique (figure 3.22).

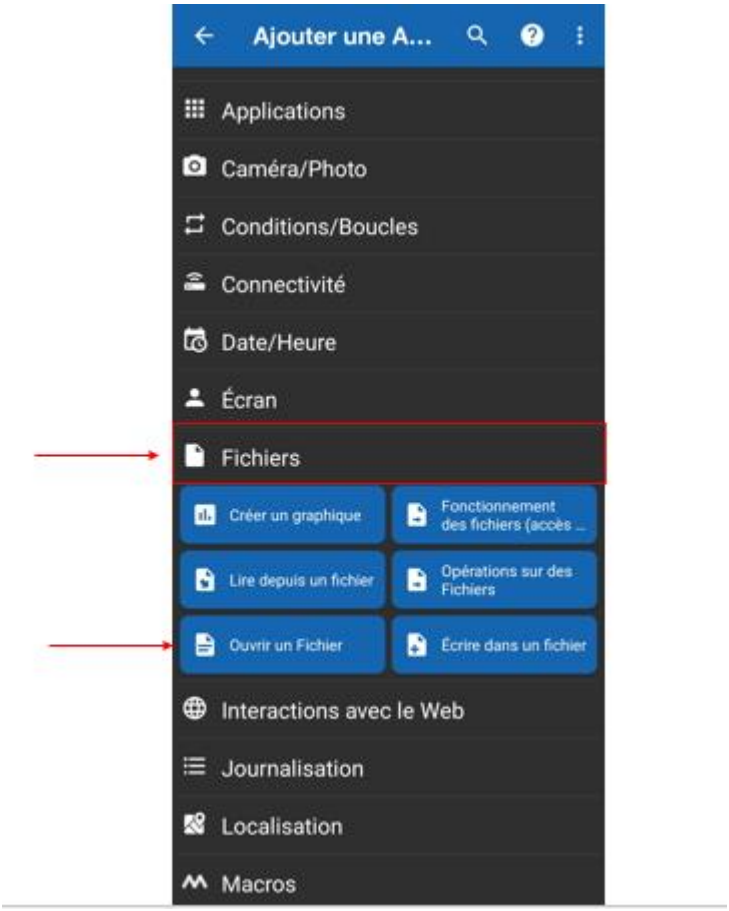


Figure 3.22. Configuration du chemin d'accès vers le stockage local.

- **Ciblage de l'image :** Dans cette étape, nous précisons la nature du fichier. Le choix de l'icône **Photo** permet au système de filtrer les fichiers afin de ne proposer que des formats d'image, tel que le code QR généré pour l'accès (figure 3.23).



Figure 3.23. Filtrage et sélection de l'image du code QR source.

- **Finalisation de la macro :** Cette étape confirme que la configuration du raccourci est terminée. Le terme **macro** désigne un ensemble d'actions (rechercher le fichier, l'ouvrir, l'afficher) regroupées en une seule commande automatique, prête à l'emploi (figure 3.24). La figure 3.25 illustre l'interface finale en mode utilisateur. Lorsque l'utilisateur appuie sur le bouton flottant configuré précédemment, le code QR d'accès s'affiche instantanément sur l'écran du smartphone (figure 3.25).



Figure 3.24. Validation finale.

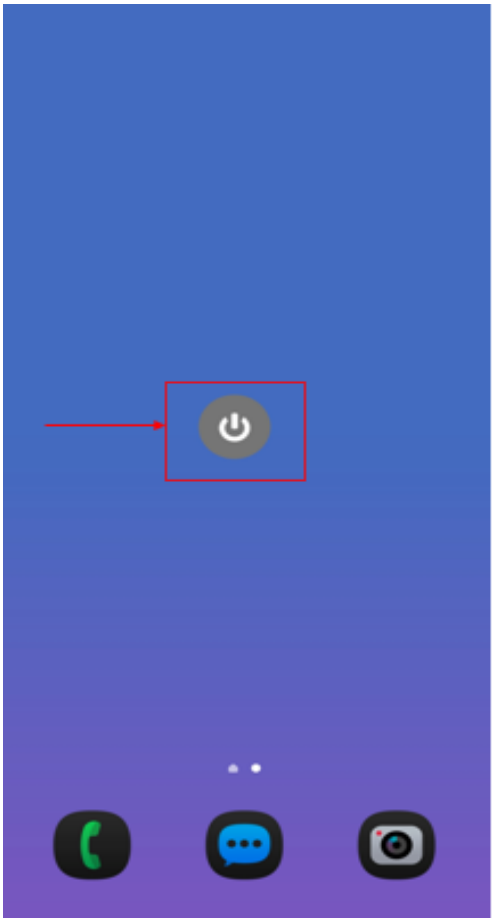


Figure 3.25. Interface utilisateur finale.

- **Utilisation pratique :** Il s'agit de l'étape durant laquelle l'utilisateur présente son écran devant la caméra ESP32-CAM de la maquette afin de déclencher le processus d'identification (figure 3.26).



Figure 3.26. Code QR utilisé.

Après avoir programmé les contrôleurs et créé le code QR, nous effectuons maintenant un petit test sur l'ESP32 pour voir s'il est prêt et comment il gère le code QR (figure 3.27).

```
QR Scanné : QR CODE VALIDE !
QR Scanné : PFE2026
```

Figure.3.27. Le succès décodage du code QR.

3.4. Description de la liaison optique et matérielle :

Il est désormais temps d'exposer l'organisation fondamentale du système d'identification et de contrôle. La figure suivante (figure 3.28) montre le montage utilisé :

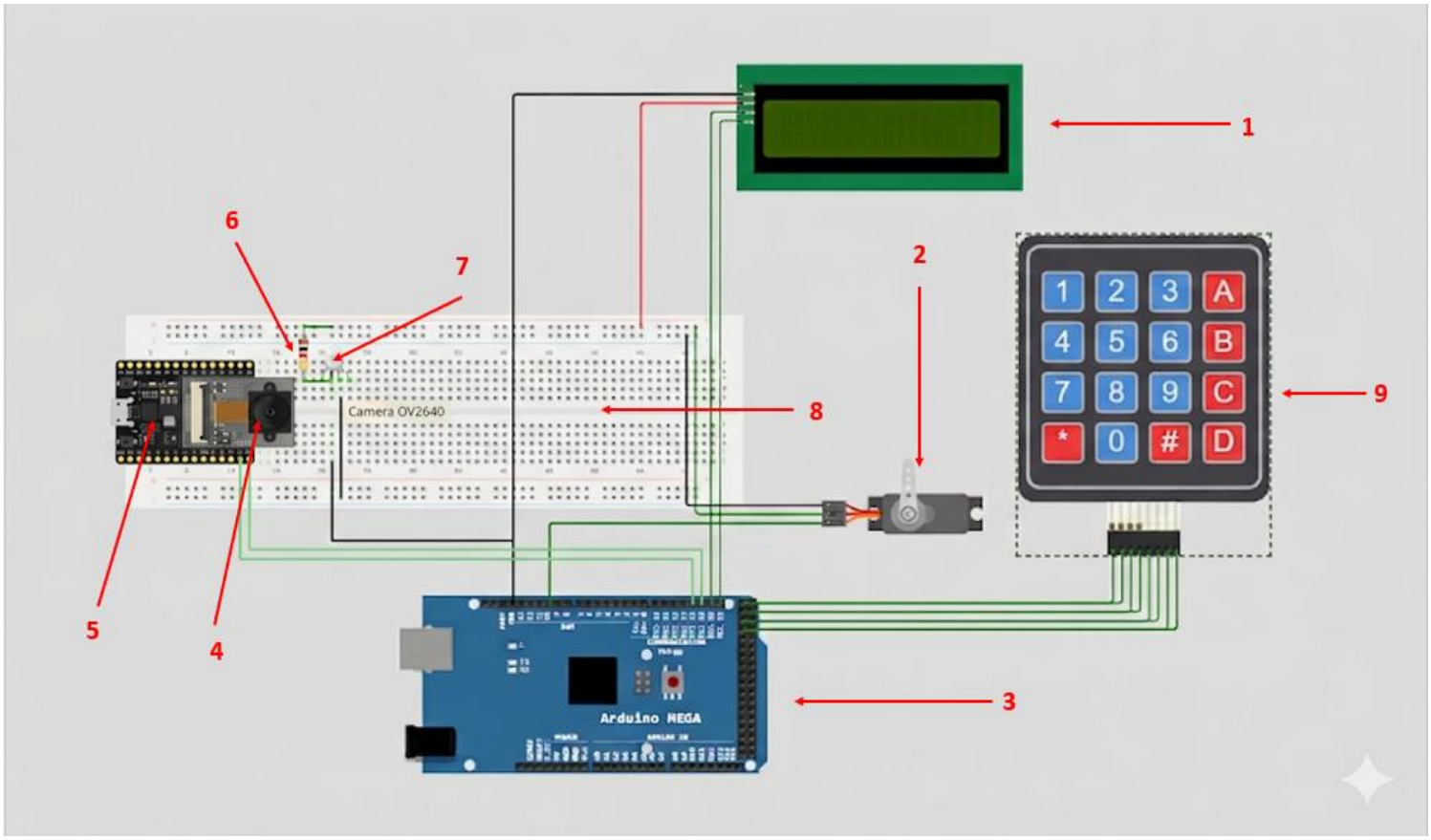


Figure.3.28. Montage de système.

Les composants utilisés dans le montage de système précédent seront présentés dans le tableau suivant (tableau 3.1).

1	Ecran LCD (16x2)
2	ServoMoteur
3	Arduino Mega 2560
4	Camera OV2640
5	Microcontrôleur ESP32 Wrover Dev
6	Resistance
7	LED (RGB)
8	Plaque D'essai
9	Clavier Matriciel

Tableau 3.1. Composants du montage.

3.4.1. Branchage des éléments :

Après avoir examiné le montage, nous vous présenterons le branchement des composants en fonction de celle-ci (tableau 3.2).

Composant	Pin sur l'Arduino Mega	Pin sur le Périphérique
Écran LCD (I2C)	20 (SDA) 21 (SCL)	SDA SCL
Clavier (Keypad)	22 à 29	Pins 1 à 8
Servomoteur	10 GND VCC (Alimentation)	Fil orange GND (-) (sur la plaque d'essai) VCC (+) (sur la plaque d'essai)
LED	2 3	Patte de la couleur verte Patte de la couleur rouge
ESP32	GND TX 18 RX 19	GND TX RX
Résistance	/	Entre la longue patte de LED et GND (-)

Tableau.3.2. Câblage des composants.

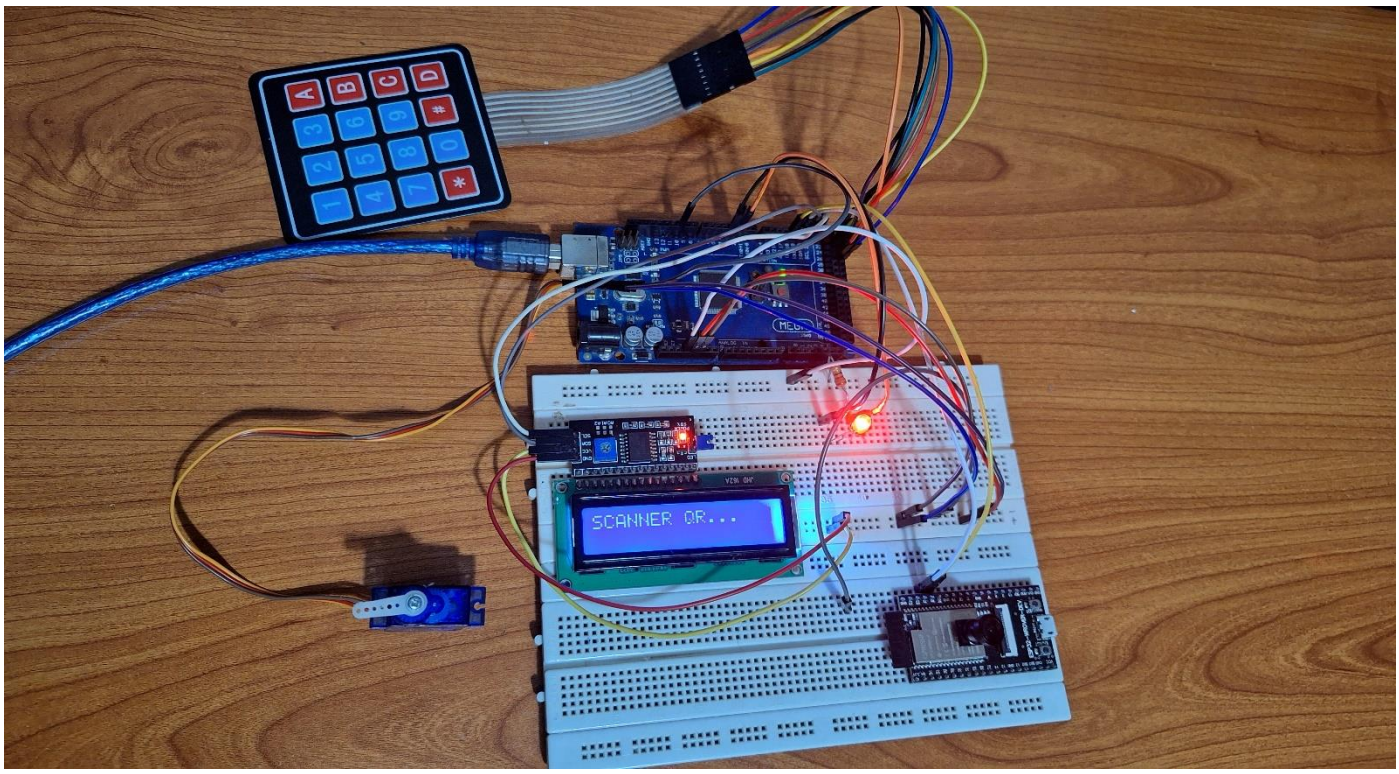


Figure.3.29. Montage réel.

Cette installation électronique forme un système intelligent destiné à l'identification et au contrôle d'accès, s'appuyant sur une communication optique unidirectionnelle permettant le décodage des codes QR. Dans cette architecture, l'Arduino Mega joue le rôle d'unité de commande principale, en supervisant et en orchestrant les divers composants, tels que l'écran LCD, le clavier matriciel et le servomoteur, qui gère l'ouverture et la fermeture de la porte.

L'Arduino Mega fonctionne en synergie avec le module ESP32-CAM, équipé de la caméra OV2640. Ce module joue le rôle de capteur optique intelligent, spécialisé dans l'acquisition, le traitement et le décodage des codes QR. Le fonctionnement repose sur la transmission optique de l'information : un smartphone émet un code QR sous forme lumineuse sur son écran, agissant ainsi comme un émetteur optique, tandis que la caméra, en tant que récepteur optique, capte cette information visuelle pour la convertir en données exploitables par le système.

Après avoir capturé l'image, l'ESP32-CAM traite ce signal optique afin d'en extraire les informations encodées dans le code QR. Ces données décodées sont ensuite transmises à l'Arduino Mega, qui se charge de vérifier l'authentification et de déclencher les actions nécessaires. Selon le résultat de cette vérification, le système autorise ou refuse l'accès, affiche les messages appropriés sur l'écran LCD et actionne le servomoteur, simulant ainsi l'ouverture ou la fermeture de la porte.

L'intégration de cette forme de communication optique offre plusieurs avantages, notamment une identification rapide, une transmission sans contact, une sécurité renforcée et une grande adaptabilité aux systèmes domotiques intelligents. En conjuguant les capacités de traitement d'image de l'ESP32-CAM pour la vision artificielle et la polyvalence de l'Arduino Mega pour la gestion des composants électroniques, le système proposé constitue une solution moderne, fiable et performante pour un contrôle d'accès intelligent.

3.5. Création d'une maquette du système d'accès sécurisé :

Afin de rendre le système plus clair et plus accessible, nous avons conçu une structure en forme de maison intégrant le montage réalisé. Ce dispositif ouvre et ferme la porte selon les étapes suivantes (figure 3.30) :



Figure.3.30. Maquette du système.

1. Au départ, le système est alimenté par une batterie externe (*power bank*) ; la LED est rouge pour indiquer que la porte est fermée, et l'écran affiche le message **Scanner QR**.
2. Ensuite, nous plaçons le code QR devant la caméra à une distance de 10 à 15 cm ; celle-ci le capture et analyse ses données en moins d'une seconde.
3. Après analyse du code QR, le système compare les données obtenues avec la valeur prédéfinie (*PFE2026*). Si elles correspondent, le message **Code correct** s'affiche à l'écran et l'utilisateur est invité à saisir le code PIN (*1234*). En revanche, si elles ne correspondent pas, le message **Code incorrect** apparaît et l'accès est refusé.

4. Une fois le message **Code correct** affiché, l'utilisateur saisit le code PIN. La porte s'ouvre alors et la LED passe du rouge au vert, confirmant ainsi la réussite de l'opération. La porte reste ouverte pendant 5 secondes, puis se referme automatiquement.

3.6. Mode de Secours Optique :

3.6.1. Principe du canal de secours :

Afin de garantir le fonctionnement de la maison intelligente même lorsque la lecture du code QR est impossible (par exemple, en cas d'écran de téléphone endommagé ou de dysfonctionnement de l'application), un canal de secours optique a été développé. Cette approche utilise la torche LED du smartphone comme émetteur lumineux, tandis que le capteur OV2640 intégré à l'ESP32-CAM assure la réception de la lumière.

3.6.2. Algorithme de décision et seuillage :

Dans un souci d'optimisation des ressources, l'algorithme n'effectue pas une analyse complète de l'image capturée. Il se limite à mesurer l'intensité lumineuse du pixel situé au centre de l'image fournie par la caméra. Cette intensité varie de 0 (noir) à 255 (blanc saturé).

Lorsque l'utilisateur dirige la torche de son smartphone vers l'objectif de la caméra, le pixel central atteint une intensité lumineuse maximale. Si cette valeur dépasse le seuil de 240 de manière continue pendant 3 secondes (afin de filtrer les perturbations lumineuses et les reflets de l'environnement) le système valide alors l'accès via le canal de secours.

3.6.3. Liaison et transmission vers l'Arduino Mega :

Une fois la condition temporelle validée, l'ESP32-CAM transmet immédiatement l'instruction `SECOURS_TORCHE_OK` à l'Arduino Mega via la liaison série UART. L'Arduino prend ensuite le relais pour afficher le message correspondant sur l'écran LCD et commander le servomoteur responsable de l'ouverture de la porte. Cette redondance logicielle confère au système une meilleure robustesse face aux défaillances potentielles (figures 3.31 et 3.32).

- **Analyse optique :**

L'ESP32-CAM mesure l'intensité lumineuse du pixel central de l'image. Si la torche du smartphone éclaire ce pixel à son intensité maximale pendant 3 secondes consécutives,

l'accès est validé et le signal SECOURS_TORCHE_OK est envoyé à l'Arduino Mega (figure 3.31).

```
100  // ======================================================
101  // LOGIQUE DE SECOURS : TORCHE CONTINUE (3 SECONDES)
102  // ======================================================
103  int centrePixel = (fb->height / 2) * fb->width + (fb->width / 2);
104  uint8_t luminanceCentrale = fb->buf[centrePixel];
105
106  if (luminanceCentrale >= SEUIL_SATURATION) {
107    if (!flashActif) {
108      debutFlash = millis();
109      flashActif = true;
110    }
111
112    if (millis() - debutFlash >= 3000) { // Condition des 3 secondes
113      Serial.println("SECOURS_TORCHE_OK"); // Envoi à l'Arduino Mega
114      vTaskDelay(3000 / portTICK_PERIOD_MS);
115      flashActif = false;
116    }
117  } else {
118    flashActif = false;
119  }
```

Figure 3.31. Partie du programme relative au mode de secours.

- **Action domotique :**

L'Arduino Mega surveille le port série. Dès réception du message SECOURS_TORCHE_OK, il efface l'écran LCD pour afficher le message ACCES SECOURS et fait pivoter le servomoteur à 90° afin de déverrouiller la porte en plexiglas.

```
55  void loop() {
56    // CAS 1 : Détection du Mode de Secours (Torche continue 3s)
57    if (message == "SECOURS_TORCHE_OK") {
58      ouvrirPorte("ACCES SECOURS", "TORCHE VALIDEE");
59    }
```

Figure 3.32. Partie du programme relative au mode de secours.

3.7. Conclusion :

En conclusion, la conception et la validation de ce système de contrôle d'accès ont permis de consolider les connaissances théoriques acquises et de développer une expertise pratique approfondie dans le domaine de la domotique. Ce chapitre a détaillé l'ensemble du

processus, depuis l'étude conceptuelle jusqu'au développement technique, aboutissant à une mise en œuvre opérationnelle répondant aux exigences de sécurité et de fiabilité fixées initialement.

Conclusion générale

Conclusion générale

Ce projet nous a permis d'étudier les différentes facettes de la maison intelligente et la manière dont elle contribue concrètement à l'amélioration de la qualité de vie. En mettant en pratique nos connaissances, nous avons pu comprendre comment l'interaction harmonieuse entre capteurs, actionneurs et plateformes de contrôle aboutit à la création d'un environnement à la fois sécurisé et interactif. Cette expérience nous a également sensibilisés aux enjeux réels de la domotique, en nous confrontant aux défis techniques et aux choix conceptuels que pose la conception d'un système embarqué fonctionnel.

Le système de contrôle d'accès que nous avons développé, intégrant la lecture de codes QR et la saisie de codes PIN, apporte une solution adaptée aux exigences de sécurité des bâtiments contemporains. Ce double niveau d'authentification — combinant une identification optique par code QR et une vérification par code PIN — renforce considérablement la fiabilité et la robustesse du système face aux tentatives d'accès non autorisées. Bien que l'investissement initial requis pour ces technologies puisse paraître important, les avantages qu'elles procurent en termes de protection des biens, d'amélioration du confort et d'autonomie pour les personnes ayant des besoins spécifiques en font une solution particulièrement précieuse. De plus, la modularité de l'architecture retenue permet d'envisager aisément des extensions futures, telles que l'ajout de nouveaux points d'accès ou l'intégration de modes d'authentification complémentaires.

En conclusion, cette démarche nous a non seulement permis d'approfondir nos connaissances techniques en matière de microcontrôleurs et de protocoles de communication, mais elle nous a aussi appris à concevoir une solution complète, de la phase de réflexion jusqu'à la réalisation concrète d'une maquette opérationnelle. Les résultats obtenus valident la faisabilité et l'efficacité de l'approche adoptée, tout en mettant en évidence certaines pistes d'amélioration, notamment en ce qui concerne la vitesse de traitement des images et la robustesse du système dans des conditions d'éclairage variables. Cette démarche ouvre ainsi des perspectives d'amélioration futures prometteuses, notamment par l'intégration de l'intelligence artificielle et des algorithmes d'apprentissage automatique afin d'optimiser la gestion de l'habitat, de renforcer la sécurité et de rendre le système plus prédictif, plus autonome et mieux adapté aux besoins de chaque utilisateur.

Références bibliographiques

Références bibliographiques

[1] Control, *Smart home ou domotique : un voyage entre différences et avantages*, 1control.eu. Disponible sur : https://www.1control.eu/blog/fr/smart-home-ou-domotique-un-voyage-entre-differences-et-avantages/.

[2] Institut de la Cognition, *Maison intelligente et domotique : l'apport des sciences cognitives*. Disponible sur : https://www.institut-cognition.com/2024/01/25/maison-intelligent-et-domotique-lapport-des-sciences-cognitives, consulté le 25/01/2024.

[3] Scribd, *Maison intelligente*. Disponible sur :

https://fr.scribd.com/document/686486970/MAISON-INTELLIGENTE.

[4] YouTube, *Vidéo de référence*. Disponible sur : https://youtu.be/3IowGFg3WvE, consulté le 21/11/2016.

[5] Computer History Museum, *The Echo IV Home Computer: 50 Years Later*. Disponible sur : https://computerhistory.org/blog/the-echo-iv-home-computer-50-years-later/, consulté le 31/05/2016.

[6] KBD News, *Honeywell H316 Kitchen Computer*. Disponible sur : https://kbd.news/Honeywell-H316-kitchen-computer-1940.html, consulté le 26/03/2025.

[7] CEA, *Essentiel sur la domotique et la maison connectée*. Disponible sur : https://www.cea.fr/comprendre/Pages/nouvelles-technologies/essentiel-sur-domotique-maison-connectee.aspx, consulté le 14/10/2016.

[8] Domadoo, *Qu'est-ce que la domotique ?* Disponible sur : https://blog.domadoo.fr/24016-quest-ce-que-la-domotique-2/, consulté le 18/06/2013.

[9] Scribd, *Chapitre 1*. Disponible sur : https://fr.scribd.com/document/551808297/chapitre-1-1.

[10] Adad Med Cherif, Bennani Nerdjes, *Smart home : défis et enjeux dans le contexte algérien*, Actes de conférence, Université de M'sila, 2024.

[11] Hello Watt, *Meilleur thermostat connecté*. Disponible sur : https://www.hellowatt.fr/domotique/thermostat-connecte/meilleur, consulté en 2026.

[12] FastCabling, *Solutions de câblage réseau et alimentation PoE pour l'habitat*. Disponible sur : https://www.fastcabling.com/fr/.

[13] Hexcell Technology, *Compteur d'énergie intelligent et unité de collecte de données DCU-P00099P1*. Disponible sur : https://fr.hexcellmeter.com/compteur-denergie-intelligent-dcu-p00099p1.html.

[14] Qivio, *Installer un détecteur de fumée chez vous*. Disponible sur : https://www.qivio.fr/habitation/blog-habitation/installer-un-detecteur-de-fumee-chez-vous-bonne-ou-mauvaise-idee, consulté le 10/10/2024.

[15] Legrand France, *Principe de mise en œuvre d'un système d'alarme intrusion*, Catalogue Technique Legrand, p. 144. Disponible sur : https://assets.legrand.com/general/legrand-fr/pfat/ca/principe_de_mise_en_oeuvre_alarme_intrusion.pdf.

[16] EntryCare, *Types of CCTV Cameras*. Disponible sur : https://entrycare.com/cctv/camera-types/, consulté en 2026.

[17] MokoSmart, *IoT Networking Technologies Guide*. Disponible sur : https://www.mokosmart.com/fr/iot-networking-technologies-guide/, consulté le 03/01/2025.

[18] Maouche Romaissa, Benchenouf Oumayma, *Étude et conception d'un contrôle domotique d'une smart home*, Mémoire de master, Université Mohamed El-Bachir El-Ibrahimi, 2024.

[19] Majumder S. et al., *Smart Homes for Elderly Healthcare — Recent Advances and Research Challenges*, Sensors, 2017.

[20] Chaib Inas, Gamri Hiba, *Conception et réalisation d'une maison intelligente à base de GSM*, Mémoire de master, Université Kasdi Merbah Ouargla, 2022.

[21] Nussey John, *Arduino pour les Nuls*, First Interactive, 2017.

[22] WayToLearnX, *Programmer l'Arduino : description de la carte Arduino*. Disponible sur : https://waytolearnx.com/2017/04/programmer-larduino-description-de-la-carte-arduino.html, consulté le 30/04/2017.

[23] A4.fr, *Carte Arduino Mega 2560 Rev3*, A4 Édition. Disponible sur : https://www.a4.fr/carte-arduino-mega-2560-rev3.html.

[24] Geekworm, *ESP32-WROVER Module Specification & Pinout*, Geekworm Wiki. Disponible sur : https://wiki.geekworm.com/ESP32-WROVER_Module, consulté le 29/04/2026.

[25] ARCELI, *Carte de développement ESP32-WROVER-DEV à faible consommation*, Amazon France. Disponible sur : https://www.amazon.fr/ARCELI-ESP32-WROVER-DEV.

[26] Kuriosity, *ESP32-CAM Camera OV2640 2MP Module Description*, Kuriosity SG. Disponible sur : https://kuriosity.sg/products/esp32-cam-camera-ov2640-2mp-1.

[27] Probots, *OV2640 160 Degree Wide Angle Lens 2MP Camera Module (24pin 21mm)*, Probots. Disponible sur : https://probots.co.in/ov2640-160-degree-wide-angle-lens-2mp-camera-module-24pin-21mm.html.

[28] Plaisir Arduino, *Afficheur LCD : comment l'exploiter avec une carte électronique*. Disponible sur : https://plaisirarduino.fr/afficheur-lcd-comment-lexploiter/, consulté le 18/04/2017.

[29] Kelai Display Technology, *How to Programme LCD Display Pins*. Disponible sur : https://www.kelaidisplay.com/how-to-programme-lcd-display-pins.html, consulté le 15/04/2025.

[30] WrayCastle, *Comprendre le fonctionnement du bus de communication I2C*. Disponible sur : https://wraycastle.com/fr/blogs/base-de-connaissances/i2c-bus, consulté le 29/11/2024.

[31] Tikno EduTech, *Module d'interface série I2C pour écran LCD*. Disponible sur : https://tikno.edutech.dz/products/module-i2c-pour-ecran-lcd-1602.

[32] Components101, *I2C Serial Interface Adapter Module Pinout & Datasheet*. Disponible sur : https://components101.com/modules/i2c-serial-interface-adapter-module, consulté le 12/06/2021.

[33] ElectronicWings, *Servo Motor Interfacing and Basics*, Electronic Wings Tutorials. Disponible sur : https://www.electronicwings.com/sensors-modules/servo-motor.

[34] EdukeyTech, *Clavier à matrice 4×4 (16 touches) pour microcontrôleurs*, Edukey Tech. Disponible sur : https://edukeytech.com/produit/keypad-clavier-matrice-4x4-16-touches/.

[35] Robotique.tech, *Tutoriel : utilisation du clavier matriciel 4×4 avec Arduino*. Disponible sur : https://www.robotique.tech/tutoriel/utilisation-du-clavier-matriciel-4x4-avec-arduino/, consulté le 08/04/2022.

[36] ViralQR, *Les différents types de codes QR et leurs applications*. Disponible sur : https://viralqr.com/fr/types-of-qr-codes/, consulté le 20/02/2025.

[37] Ionos, *Qu'est-ce qu'un QR code ? Origine et fonctionnement du code à deux dimensions*. Disponible sur : https://www.ionos.fr/digitalguide/web-marketing/vendre-sur-internet/quest-ce-quun-qr-code/, consulté le 17/10/2022.

[38] Wikipédia, *Code QR : historique, structure et encodage des données*. Disponible sur : https://fr.wikipedia.org/wiki/Code_QR.

[39] Arduino Blaise Pascal, *Présentation et prise en main du logiciel Arduino IDE*, Lycée Blaise Pascal. Disponible sur : https://arduino.blaisepascal.fr/presentation/logiciel/, consulté le 20/12/2020.

[40] Bio4ol Blog, *MacroDroid ou comment automatiser des tâches complexes sur son smartphone*. Disponible sur : https://bi4ol.yj.fr/blog/macrodroid-ou-comment-automatiser-des-taches-sur-smartphone/.

[41] Scribd, *Guide comparatif : les différents types de cartes Arduino et leurs caractéristiques*. Disponible sur : https://fr.scribd.com/document/703509409/differents-types-d-arduino.

[42] Beldjazia Djihane, Bouhlas Widad, *Amélioration des performances énergétiques d'une maison intelligente*, Mémoire de master, Université de Jijel, 2022.