



**Département de Physique  
Laboratoire de Traitement de l'Information**

**Mémoire de Projet de Fin d'Études**

**Filière Sciences de la Matière Physique**

**Parcours : Électronique**

**Sujet :**

**Conception et Réalisation d'une Serrure  
Électronique Intelligente en utilisant ESP32 et  
RFID**

Année Universitaire 2024-2025



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ  
الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ  
الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ



## Liste des figures

Figure 1 : Serrure connectée en RFID .....	6
Figure 2 : fonctionne de Serrure connectée en RFID .....	7
Figure 3 : Serrures connectées en NFC .....	8
Figure 4 : serrure connectée en Bluetooth .....	9
Figure 5 : Serrure connectée en WIFI .....	10
Figure 6 : Serrure connectée à commande vocale .....	11
Figure 7 : serrures à code .....	12
Figure 8 : Serrure à infrarouge .....	13
Figure 9 : Serrures biométriques .....	14
Figure 10 : ESP32 WROOM 32 GPIO Pins .....	18
Figure 11 : ESP32 WROOM 32 .....	18
Figure 12 : Schéma simplifié de la carte ESP32-WROOM-32S .....	19
Figure 13 : module RC522 RFID .....	21
Figure 14 : schéma d'un RFID-RC522 .....	21
Figure 15 : schéma d'un tag RFID .....	23
Figure 16 : schéma d'une étiquette RFID .....	23
Figure 17 : schéma de fonctionnement RFID .....	24
Figure 18 : clavier matriciel .....	24
Figure 19 : schéma de connexion d'un clavier matriciel .....	25
Figure 20 : LCD I2C .....	25
Figure 21 : schéma de fonctionnement de LED .....	26
Figure 22 : LED .....	26
Figure 23 : schéma de fonctionnement de buzzer .....	27
Figure 24 : Buzzer électronique .....	27
Figure 25 : Serrure électromagnétique .....	28
Figure 26 : Caractéristiques Techniques de serrure électromagnétique .....	29
Figure 27 : Relais .....	30
Figure 28 : Application Blynk .....	31
Figure 29 : programme wokwi .....	34
Figure 30 : Alimentation d'esp32 .....	35
Figure 31 : câblage du keypad .....	35
Figure 32 : câblage de LCD I2C .....	36
Figure 33 : câblage de RFID .....	37
Figure 34 : câblage des LED .....	37
Figure 35 : connexion le buzzer avec l'esp32 .....	38
Figure 36 : connexion le relais avec l'esp32 .....	38
Figure 37 : connexion de la serrure avec l'esp32 .....	39
Figure 38 : schéma complet de système .....	39
Figure 39 : Organigramme représentant le fonctionnement du système .....	40
Figure 40 : la fenêtre d'Arduino IDE .....	41
Figure 41 : Programmation d'ESP32 déclaration des librairies .....	43
Figure 42 : Programmation d'ESP32 Définitions des broches .....	43
Figure 43 : Programmation d'ESP32 initialisations de clavier .....	43
Figure 44 : Programmation d'ESP32 : déclaration des variables .....	44
Figure 45 : Programmation d'ESP32 Information d'identification .....	44

Figure 46 : Programmation d'ESP32 fonctions utilisées .....	44
Figure 47 : Programmation d'ESP32 Activation de WIFI.....	45
Figure 48 : Écran de création d'un nouveau modèle sur la Blynk Console.....	45
Figure 49 : Page de configuration des Datastreams dans la Blynk Console .....	46
Figure 50 : Fenêtre de configuration d'un Virtual Pin Datastream. ....	46
Figure 51 : L'interface Blynk avant la connexion de l'ESP32, montrant l'état hors ligne.....	47
Figure 52 : L'état "on et off" du bouton dans Blynk .....	48
Figure 53 : Création d'une feuille Google Sheets .....	48
Figure 54 : code Google Apps Script .....	49
Figure 55 : Publication du script .....	49
Figure 56 : montage global.....	50
Figure 57 : Accès par badge RFID .....	50
Figure 58 : Accès par code PIN.....	51
Figure 59 : Accès par point d'accès Wi-Fi .....	51
Figure 60 : Accès via application Blynk .....	52
Figure 61 : Résultats dans Google Sheets .....	52
Figure 62 : Vue frontale du prototype .....	53
Figure 63 : Vue arrière du prototype .....	53

# Sommaire

Remerciements .....	<b>Erreur ! Signet non défini.</b>
Liste des figures.....	VI
Sommaire.....	VIII
Résumé .....	X
Introduction Générale .....	2
Chapitre I : Généralité sur les serrures électroniques	
1. Introduction.....	5
2. Serrures électroniques .....	5
3. Types des serrures électroniques.....	6
3.1. Serrures intelligentes .....	6
3.1.1. Serrures connectées en RFID .....	6
3.1.2. Serrures connectées en NFC.....	7
3.1.3. Serrures connectées en Bluetooth.....	8
3.1.4. Serrures connectées en WIFI.....	10
3.1.5. Serrure connectée à commande vocale .....	10
3.2 Serrures non intelligentes .....	11
3.2.1. Serrure à code PIN .....	11
3.2.2. Serrures à infrarouge (IR).....	12
3.2.3. Serrures biométriques.....	14
4. Cahier des charges .....	14
4.1. Exigences Fonctionnelles .....	14
4.2. Exigences Techniques .....	15
5. Conclusion .....	16
Chapitre II : Présentation des composants du système étudié	
1. Introduction .....	18
2. Unité de commande est de traitement.....	18
2.1. ESP32 .....	18
3. Unité communication.....	20
3.1. Périphérique d'entrée .....	20
3.1.1. RFID (Identification par Radiofréquence) .....	20
3.1.1.1. Lecteur de carte RFID .....	21
3.1.1.2. Tag (badge) RFID .....	22
3.1.1.3. Fonctionnement de la communication de RFID .....	23
3.1.2. Clavier matriciel 4x4 .....	24
3.2. Périphérique de sortie .....	25
3.2.1. Afficheur LCD I2C .....	25
3.2.2. Diode électroluminescente LED .....	26
3.2.3. Buzzer électronique .....	27
4. Unité des Actionneurs .....	28
4.1. Serrure électromagnétique .....	28
4.2. Module Relais 5v.....	29
5. Application mobile.....	30
5.1. Application Blynk.....	30
5.2. Communication entre ESP32 et application mobile .....	32
6. Conclusion .....	32
Chapitre III : Simulation et Réalisation du Système	
1. Introduction.....	34
2. Simulation du système .....	34

---

2.1. Programme wokwi.....	34
2.2. Câblage des composants à l'esp32 .....	35
2.2.1. Alimentation d'esp32 .....	35
2.2.2. Clavier matriciel 4x4 .....	35
2.2.3. LCD I2C .....	36
2.2.4. RFID RC522.....	36
2.2.5. Câblage des LED .....	37
2.2.6. Buzzer.....	38
2.2.7. Relais .....	38
2.2.8. Connexion d'un Serrure électromagnétique à l'esp32 .....	39
2.2.9. Schéma complet du système.....	39
3. Organigramme du système.....	40
4. Réalisation du système.....	41
4.1. Réalisation logicielle .....	41
4.1.1. Arduino IDE .....	41
4.1.2. Déroulement du programme.....	42
4.1.3. Partie déclaration.....	43
4.1.4. Activation de wifi .....	44
4.1.5. Activation de Blynk avec esp32.....	45
4.1.6. Intégration de Google Sheets .....	48
4.1.6.1. Création d'une feuille Google Sheets .....	48
4.1.6.2. Écriture du Script Google Apps .....	49
4.1.6.3. Publication du script en tant que Web App.....	49
4.2. Réalisation physique .....	50
4.2.1. Montage global.....	50
4.2.2. Résultats pratiques.....	50
4.2.3. Présentation finale du système .....	53
5. Conclusion .....	54
Conclusion Générale .....	56
Annexe.....	58
Références .....	68

## Résumé

Ce projet vise à concevoir et développer un système de serrure électronique, représentant une solution moderne et sécurisée pour le contrôle d'accès. Le rapport débute par une introduction générale aux serrures électroniques et à leurs différents types, abordant les distinctions entre les serrures intelligentes (connectées via des technologies telles que RFID, NFC, Bluetooth, Wi-Fi et commande vocale) et les serrures non intelligentes (comme les serrures à code PIN, infrarouge et biométriques). Le projet définit des exigences fonctionnelles et techniques précises pour le système proposé, garantissant ainsi la réalisation des objectifs en termes de sécurité et de facilité d'utilisation. Le rapport détaille les composants matériels et logiciels utilisés dans le système, notamment la carte ESP32, le lecteur RFID, le serrure électromagnétique et les autres composants essentiels au fonctionnement. Il explique également les mécanismes de communication et de contrôle entre ces composants. Le rapport inclut également une simulation du système et examine comment les différents composants interagissent pour réaliser la fonction de serrure électronique. Le projet se conclut par une discussion des résultats et des recommandations futures, soulignant le potentiel de développement du système pour améliorer les performances et la sécurité. Globalement, ce projet présente un modèle pratique d'application des technologies de serrures électroniques, en mettant l'accent sur l'intégration du matériel et du logiciel pour obtenir un système de contrôle d'accès efficace et sécurisé.

## Abstract

This project aims to design and develop an electronic lock system, representing a modern and secure solution for access control. The report begins with a general introduction to electronic locks and their various types, addressing the distinctions between smart locks (connected via technologies such as RFID, NFC, Bluetooth, Wi-Fi, and voice control) and non-smart locks (such as PIN code, infrared, and biometric locks). The project defines precise functional and technical requirements for the proposed system, ensuring that the desired objectives in terms of security and ease of use are met. The report elaborates on the hardware and software components used in the system, including the ESP32 board, RFID reader, electromagnetic lock, and other essential components for operation. It also explains the communication and control

mechanisms between these components. The report also includes a system simulation and reviews how the various components interact to achieve the electronic lock function. The project concludes with a discussion of the results and future recommendations, highlighting the system's development potential for improved performance and security. Overall, this project presents a practical model for implementing electronic lock technologies, focusing on the integration of hardware and software to achieve an effective and secure access control system.

# Introduction Générale

## Introduction Générale

L'être humain cherche toujours à mettre en place un système de sécurité et de surveillance fiable afin de protéger ses biens immobilier et les locaux collectifs contre les intrusions et les prévenir contre le vol. Les serrures ont pour but d'assurer cette tâche depuis longtemps et ne cesse pas à évoluer jusqu'au nos jours dont on trouve des serrures dites intelligentes permet de gérer l'accès aux endroits privés d'une manière très pratique. L'évolution technologique a permis le développement des systèmes de sécurité qui deviennent de plus en plus performants. Cette évolution est due essentiellement à l'utilisation des applications de l'électronique moderne du point de vue communication entre les périphériques de commande (Bluetooth, WIFI, Infra rouge...) et côté composants (microcontrôleurs programmables, carte ARDUINO..).

Avec l'évolution des technologies, les méthodes traditionnelles de sécurisation des maisons et des locaux ne répondent plus toujours aux besoins actuels. Les pertes de clés, le manque de flexibilité, et l'absence de contrôle à distance posent souvent des problèmes aux utilisateurs. Cela a créé un besoin croissant de solutions modernes, pratiques et sécurisées, capables d'offrir plusieurs options d'accès adaptées aux nouvelles exigences du quotidien.

À travers ce projet, notre objectif est de concevoir un verrou intelligent basé sur une carte ESP32, qui peut être utilisé comme solution pratique et sécurisée dans le domaine de la protection.

L'idée, c'est de faciliter l'accès tout en offrant plusieurs moyens d'ouverture, au lieu d'utiliser uniquement une clé classique.

La serrure électronique que nous souhaitons réaliser pourra être ouverte de quatre manières différentes :

Par une carte RFID,

En saisissant un code PIN,

Via l'application Blynk connectée en Wi-Fi,

Ou bien en passant par un point d'accès local créé par la carte elle-même.

L'objectif n'est donc pas seulement de fabriquer un simple verrou électronique, mais de proposer un système intelligent, facile à utiliser, fiable, et capable de s'adapter aux besoins de sécurité du quotidien.

Au début, nous allons étudier les différents composants du projet, comme la carte ESP32, le lecteur RFID, le clavier pour le code PIN, ainsi que la connexion Wi-Fi pour l'application Blynk.

Pour valider notre idée, nous allons utiliser le logiciel Wokwi afin de simuler le circuit électronique et programmer notre code. Cette simulation nous a permis de tester tout le système dans un environnement virtuel, ce qui nous a évité des erreurs lors de la réalisation matérielle.

Une fois que nous avons confirmé que tout fonctionnait correctement en simulation, nous sommes passés à la réalisation pratique. Nous avons assemblé les composants et les avons connectés à la carte ESP32.

Nous allons programmer le système pour contrôler le verrou par quatre méthodes différentes : RFID, code PIN, application Blynk via Wi-Fi, et point d'accès local.

Chaque méthode a été testée séparément, puis combinée pour vérifier la stabilité et la facilité d'utilisation du système.

Pour finir, ce projet nous a permis de créer un verrou intelligent facile à utiliser, sûr, et qui fonctionne bien avec plusieurs méthodes d'ouverture.

# Chapitre I : Généralité sur les serrures électroniques

## Chapitre I : Généralité sur les serrures électroniques

### 1. Introduction

La serrure électronique, c'est l'évolution technologique appliquée à votre sécurité et à votre confort d'utilisation, permet d'ouvrir et de fermer un objet comme une porte sans une clé. On a besoin d'un dispositif qui nécessite simplement un courant électrique pour verrouiller ou déverrouiller une porte. Elle peut être installée sur une porte d'immeuble, sur une porte d'une salle blanche, mais également sur la porte d'une chambre d'hôtel ou encore d'une voiture. Il existe plusieurs types de serrures électroniques pour les portes, Elles se distinguent selon le système de verrouillage :

- Les serrures connectées Bluetooth
- Les serrures à smart code.
- Les serrures badge (RFID)
- Les serrures connectées WIFI.
- Les serrures biométriques.
- Les serrures à télécommandes IR
- les serrures infrarouges (IR)

### 2. Serrures électroniques

La serrure électronique est un dispositif de verrouillage qui fonctionne au moyen d'un courant électrique, et aussi c'est un système de fermeture d'une porte ou d'une fenêtre qui ne peut être ouvert qu'à l'aide d'une clé ou d'un bouton ou à l'aide de moyens électroniques avancé. [1]

Une serrure connectée fonctionne grâce à une combinaison de composants électroniques et de réseaux de communication. Elle est généralement dotée d'un mécanisme de verrouillage électronique, d'un système de contrôle à distance et d'une interface utilisateur via une application mobile ou un dispositif dédié. [2]

Les serrures électroniques offrent une gamme d'avantages qui les rendent attrayantes pour les utilisateurs résidentiels et professionnels, tels que :

- Une sécurité renforcée grâce à la difficulté de les forcer sans le bon signal électrique.
- La possibilité d'intégration dans des systèmes de contrôle d'accès pour une gestion facile des entrées.

- Plus de confort d'utilisation, avec des options telles que l'ouverture à distance ou par empreinte digitale. [3]

### 3. Types des serrures électroniques

#### 3.1. Serrures intelligentes

##### 3.1.1. Serrures connectées en RFID

Les racines de la technologie d'identification par radiofréquence (RFID) remontent à la Seconde Guerre mondiale, où elle était principalement utilisée pour distinguer les amis des ennemis. [4]

Au cours des années 1970, une étape cruciale dans l'évolution de la technologie RFID a été franchie avec le développement du premier émetteur-récepteur sans fil doté d'une étiquette passive et d'une mémoire. Cette innovation a été l'œuvre de Mario Cardullo, un inventeur visionnaire qui a compris le potentiel de l'identification à distance pour diverses applications. À l'époque, cette technologie a été envisagée et utilisée principalement à des fins douanières, marquant ainsi les premiers pas de la RFID dans le domaine commercial. [5]

Le mécanisme de fonctionnement de cette technologie RFID, qui se distingue par un ensemble de propriétés uniques telles que l'automatisation, la sécurité et l'efficacité, permettra des utilisations plus larges englobant les domaines de la fabrication, des chaînes d'approvisionnement, de l'agriculture, du transport, de la santé et des services. [6]

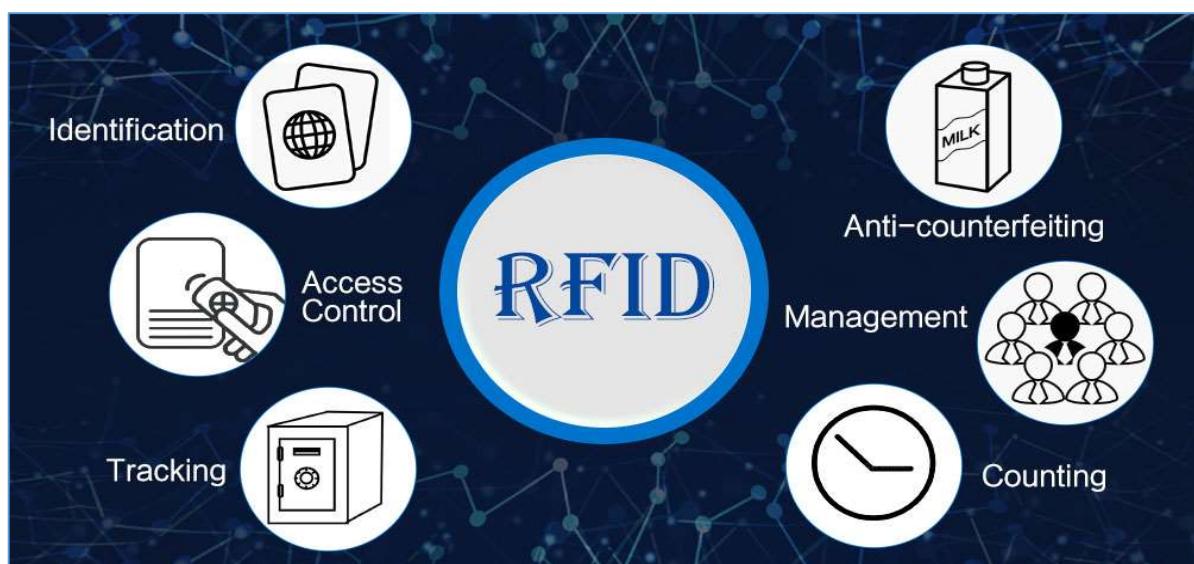


Figure 1 : Serrure connectée en RFID

Un système de verrouillage RFID se compose de deux éléments essentiels : le lecteur et l'étiquette, qui permettent l'échange de données d'identification.

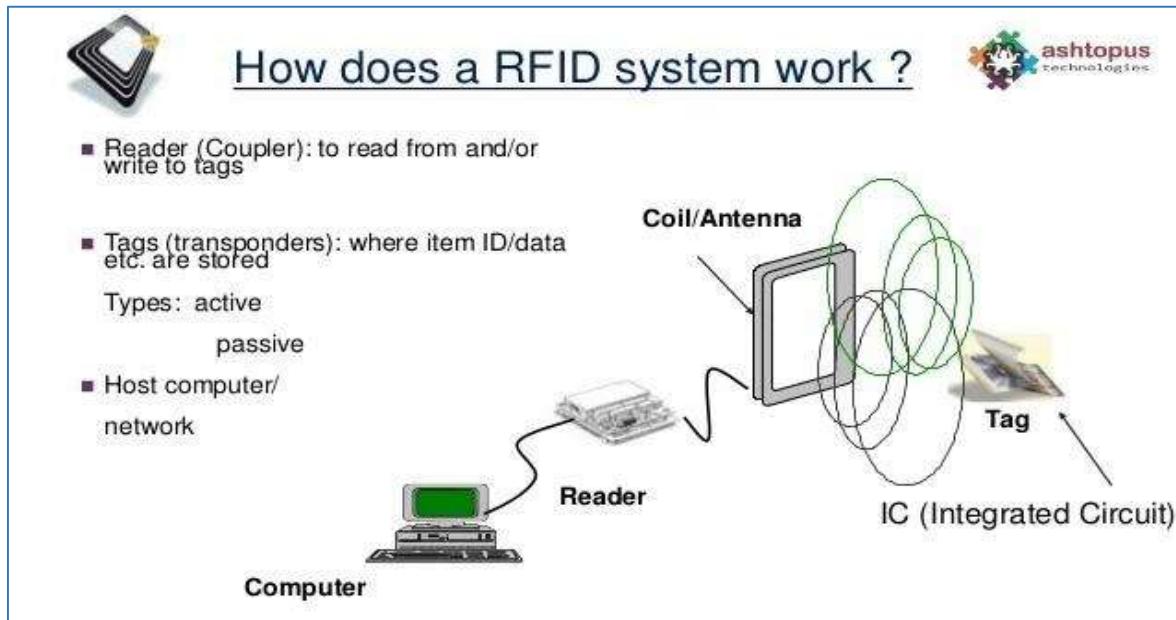


Figure 2 : fonctionne de Serrure connectée en RFID

Ce processus se déroule en trois étapes principales. Premièrement, le lecteur émet des ondes radio à une fréquence spécifique, générant ainsi un champ électromagnétique. Deuxièmement, l'antenne de l'étiquette capte l'énergie de ces ondes radio. (Dans le cas des étiquettes passives, cette énergie est utilisée pour alimenter la puce électronique.) Cela stimule la puce de l'étiquette à transmettre les données stockées via son antenne. (Les ondes radio émises sont modulées pour encoder les données.).

Troisièmement, le lecteur reçoit les ondes radio renvoyées par l'étiquette, les démodule pour extraire les informations. Enfin, le lecteur traite les données et les transfère vers un système informatique ou une base de données. [7]

### 3.1.2. Serrures connectées en NFC

La technologie NFC (Near Field communication) est une technologie sans fil de courte portée qui permet l'échange d'informations entre périphériques à une distance généralement inférieure à 10 cm.

Elle repose sur la communication via des ondes radio à haute fréquence (13,56 MHz) Et aussi permet une interaction sécurisée et rapide entre deux appareils, comme des smartphone, des terminaux de paiement. [8]

Son fonctionnement s'appuie sur l'induction électromagnétique entre deux dispositifs, lorsque l'un, généralement un smartphone ou une carte NFC, passe à proximité d'un lecteur, en l'occurrence la serrure. Cette interaction permet de déverrouiller ou verrouiller une porte de manière sécurisée et rapide.



Figure 3 : Serrures connectées en NFC

Les avantages de la serrure NFC pour la sécurité du domicile :

Une serrure NFC offre de multiples avantages pour la sécurisation domestique. Tout d'abord, elle supprime le besoin de clés physiques, réduisant ainsi les risques de perte ou de vol. De plus, elle permet un contrôle d'accès personnalisé : des clés virtuelles peuvent être attribuées et révoquées à tout moment via une application mobile. L'authentification par NFC est également très difficile à pirater, car elle nécessite une proximité physique avec la serrure, ce qui renforce la protection contre les éventuelles effractions. [9]

### 3.1.3. Serrures connectées en Bluetooth

Bluetooth est une technologie de communication sans fil à courte portée développée en 1994 par Ericsson. Elle fonctionne à des fréquences comprises entre 2,4 et 2,483 GHz, avec des débits allant jusqu'à 1 Mbit/s. L'un de ses avantages est qu'il élimine le besoin d'utiliser des câbles entre des appareils tels que les smartphones, les ordinateurs, les imprimantes et les claviers, ce qui facilite leur connexion sans avoir besoin de fils. [10]

Une serrure connectée en Bluetooth est un appareil intelligent qui utilise la connectivité sans fil avec les téléphones ou les tablettes, permettant aux utilisateurs d'utiliser leurs smartphones comme clé.

Cette serrure permet également à la porte de s'ouvrir automatiquement lorsque vous vous en approchez et de se fermer lorsque vous partez, grâce à la technologie Bluetooth de votre smartphone.



Figure 4 : serrure connectée en Bluetooth

Cette serrure utilise la technologie BLE (Bluetooth Low Energy), qui fournit une connexion stable avec une faible consommation d'énergie, et est contrôlée via une application pour smartphone, qui permet à l'administrateur de configurer les autorisations et de définir quels utilisateurs sont autorisés à entrer.

Lorsqu'une personne autorisée s'approche de la zone de couverture Bluetooth, l'application envoie un signal chiffré à la serrure, qui vérifie l'identité de la personne et déverrouille automatiquement la porte. La serrure se verrouille ensuite dès que la personne quitte la zone de couverture Bluetooth. [11]

Les serrures intelligentes Bluetooth offrent un niveau élevé de sécurité et de commodité, ainsi qu'une facilité d'utilisation. Contrairement aux serrures traditionnelles, elle repose sur une fonction d'ouverture automatique lorsque l'utilisateur s'approche.

Ce type de serrure permet également de créer et de gérer des automatismes d'accès individuels à votre domicile grâce à une application disponible sur les smartphones fonctionnant sous Android ou IOS (système d'exploitation mobile d'Apple). Cette application dispose d'un journal d'activité 24h/24 et 7j/7. [12]

### 3.1.4. Serrures connectées en WIFI

‘Wireless Fidelity’ qui peut être traduite en français par "fidélité sans fil", Apparues pour la première fois en 1997 est un ensemble de protocoles de communication sans fil régi par les normes du groupe IEEE 802.11 (ISO/CEI 8802-11). Un réseau Wi-Fi permet de relier par ondes radio plusieurs appareils informatiques (ordinateur, routeur, smartphone, , modem Internet, etc.)



Figure 5 : Serrure connectée en WIFI

Fonctionnant via des applications, ces produits vous permettent de verrouiller ou de déverrouiller votre porte de n'importe où avec une connexion sans fil. Vous pouvez également suivre l'historique d'ouverture et de fermeture et partager des clés électroniques avec la famille, les voisins, ext.

### 3.1.5. Serrure connectée à commande vocale

Une serrure à commande vocale est une serrure intelligente qui s'ouvre et se ferme grâce aux commandes vocales de l'utilisateur, plutôt qu'avec des clés, des empreintes digitales ou un code. Elle s'appuie sur l'intelligence artificielle et la reconnaissance vocale pour distinguer et identifier les commandes de l'utilisateur.

Cette serrure fonctionne au moyen d'un microphone qui reçoit des commandes vocales, qui sont ensuite analysées par un programme d'intelligence artificielle pour déterminer si la commande vocale correspond aux commandes précédemment programmées.



Figure 6 : Serrure connectée à commande vocale

Après la correspondance, la serrure envoie un signal électronique pour ouvrir ou fermer la serrure. Dans certaines serrures, la reconnaissance des caractéristiques de la voix de l'utilisateur (telles que le ton de la voix, la fréquence) est utilisée.

Avantages de cette serrure :

- Haute sécurité
- Facilité d'utilisation
- Contrôle mains libres
- Ouvrez la porte avec des commandes simples.
- Possibilité de le connecter à des assistants intelligents tels que Google Assistant, Alexa ou Apple HomeKit.

## 3.2. Serrures non intelligentes

### 3.2.1. Serrure à code PIN

La serrure à code est un dispositif de sécurité innovant qui remplace la clé traditionnelle par un code numérique. Pour déverrouiller la serrure, l'utilisateur saisit un code sur un clavier intégré. Si le code correspond à la combinaison préenregistrée, le mécanisme de verrouillage s'active, permettant l'ouverture de la porte ou de l'objet protégé.

Les serrures à code sont couramment utilisées dans divers contextes, tels que :

Portes d'entrée : elles peuvent être très pratiques notamment pour sécuriser l'accès à des bureaux, à un immeuble ou à tout autre bâtiment où plusieurs personnes doivent avoir accès sans avoir à se soucier des clés.

Coffres-forts : les serrures à code sont souvent utilisées pour sécuriser des coffres-forts, des armoires de sécurité ou des casiers.

Véhicules : certains modèles de voitures et de véhicules utilitaires sont équipés de serrures à code pour permettre l'accès sans clé.

Valises et bagages : certaines valises et sacs de voyage sont équipés de serrures à code pour protéger les biens personnels des voyageurs.



Figure 7 : serrures à code

La serrure à code électronique est un système de verrouillage sophistiqué qui utilise un clavier et des circuits électroniques pour stocker et vérifier un code d'accès. Lorsqu'un code valide est saisi, un signal électronique active le mécanisme de déverrouillage, permettant l'ouverture de la porte ou de l'objet protégé

Ce genre de serrure présente plusieurs avantages :

- Simplicité et tranquillité d'esprit : « Dites adieu aux clés perdues grâce à la serrure à code. Plus besoin de craindre de rester enfermé dehors ou de faire appel à un serrurier en urgence
- Flexibilité du code : Dans de nombreux modèles de serrures à code, il est possible de changer le code à tout moment. Cela permet de modifier régulièrement le code pour renforcer la sécurité
- Contrôle d'accès : Les serrures à code permettent de contrôler facilement l'accès à une zone restreinte. Vous pouvez partager le code avec les personnes autorisées et révoquer l'accès en modifiant simplement le code lorsque nécessaire.

### 3.2.2. Serrures à infrarouge (IR)

Une serrure à infrarouge est un dispositif de verrouillage électronique qui intègre des capteurs infrarouges pour détecter la présence ou l'absence d'un individu à proximité de la porte. Ces capteurs réagissent aux variations du rayonnement infrarouge émis par le corps humain en mouvement, permettant ainsi de contrôler l'accès sans contact physique direct. [13]

En somme, une serrure à infrarouge combine des technologies de détection sans contact avec des mécanismes de verrouillage automatisés, offrant ainsi une solution sécurisée et pratique pour le contrôle d'accès.

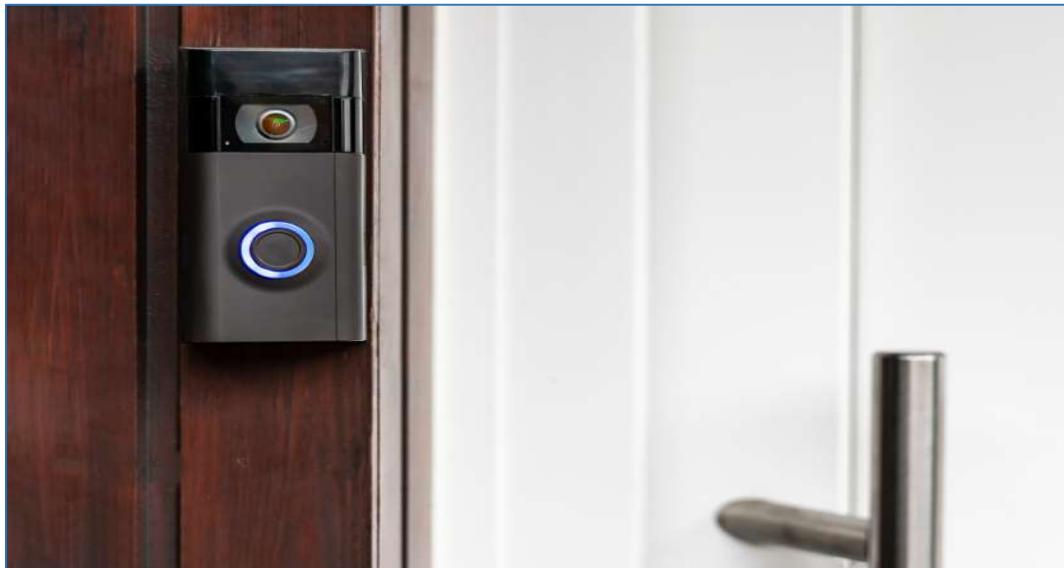


Figure 8 : Serrure à infrarouge

Une serrure à infrarouge fonctionne généralement à l'aide de capteurs qui détectent un signal infrarouge émis par une télécommande, un badge, ou un autre dispositif de déverrouillage. Le principe repose sur la communication sans fil entre la serrure et un émetteur.

- Émission du signal : Un dispositif émetteur envoie un signal infrarouge à la serrure lorsqu'il est activé
- Détection du signal : La serrure est équipée d'un récepteur infrarouge qui capte ce signal. Ce récepteur détecte la fréquence spécifique ou le code du signal infrarouge émis par l'émetteur.
- Identification et validation : Une fois le signal capté, le récepteur de la serrure analyse le signal pour vérifier s'il correspond à un signal autorisé
- Déverrouillage : Si le signal est valide, le système interne de la serrure déclenche un mécanisme qui permet d'ouvrir la porte.

Les serrures à infrarouge sont populaires pour leur sécurité et leur praticité, car elles n'ont pas besoin de contact physique direct (comme une clé) et sont souvent utilisées dans des environnements nécessitant un contrôle d'accès sans contact, comme dans les hôtels ou certains bâtiments sécurisés. [14]

### 3.2.3. Serrures biométriques

Une serrure biométrique est un système de gestion d'accès qui utilise la reconnaissance des empreintes digitales ou de l'iris. Seules les personnes enregistrées peuvent déverrouiller.



Figure 9 : Serrures biométriques

Une serrure biométrique fonctionne grâce à un capteur capable de lire les empreintes digitales ou l'iris. Le système traite ensuite les données et recherche des correspondances dans sa base de données. Si une correspondance est trouvée avec les caractéristiques morphologiques enregistrées de la personne, le verrou est ouvert.

Les avantages de cette serrure incluent le fait qu'elle offre un niveau de sécurité élevé et rend l'entrée plus facile et plus rapide sans avoir besoin de clés traditionnelles. De plus, il peut être intégré aux systèmes de maison intelligente, ce qui permet de le contrôler à distance via une application pour smartphone. [15]

## 4. Cahier des charges

### 4.1. Exigences Fonctionnelles

Ce projet vise à concevoir une serrure électronique intelligente basée sur plusieurs méthodes d'authentification, garantissant à la fois sécurité et facilité d'utilisation. La serrure doit offrir les fonctionnalités suivantes :

Ouverture de la serrure via :

- Un code PIN saisi sur un clavier (Keypad).
- Une carte RFID pour vérifier l'identité de l'utilisateur.
- Une application mobile pour un contrôle à distance.

- Une connexion Wi-Fi pour une gestion à distance via Internet.

Fermeture automatique de la serrure après une période d'inactivité.

Envoi de notifications à l'application en cas de plusieurs tentatives de code erroné ou de tentative d'intrusion.

Enregistrement des entrées et sorties via l'application, avec possibilité de consulter l'historique d'utilisation.

Possibilité d'ajouter ou de supprimer des utilisateurs via l'application.

## 4.2. Exigences Techniques

Pour assurer ces fonctionnalités, les composants et technologies suivants seront utilisés :

Unité de contrôle électronique :

- Utilisation de l'ESP32 pour sa capacité à prendre en charge simultanément le WiFi et le Bluetooth.

Composants de la serrure :

- Clavier (Keypad 4x4) pour la saisie du code PIN.
- Module de lecture RFID (RC522) pour vérifier les cartes autorisées.
- Module Wi-Fi/Bluetooth (intégré à l'ESP32) pour permettre le contrôle à distance.
- Moteur électrique (Servo ou Solenoid Lock) pour gérer l'ouverture et la fermeture de la serrure.

Alimentation électrique :

- Fonctionnement sur batterie rechargeable ou via un adaptateur secteur (5V-12V).
- Faible consommation d'énergie pour garantir une autonomie prolongée.

Programmation et développement :

- Programmation de la serrure en python
- l'IDE Arduino.
- Développement de l'application mobile avec Flutter ou Android Studio, prenant en charge la connexion WiFi.
- Mise à jour du programme via WiFi (OTA Update) pour ajouter de nouvelles fonctionnalités ou renforcer la sécurité.

## 5. Conclusion

Dans ce chapitre, nous avons présenté différents types de serrures électroniques avec une explication des diverses technologies qu'elles utilisent, ce qui renforce le niveau de sécurité contre les tentatives de piratage et d'intrusion.

De plus, certaines serrures électroniques supportent plusieurs méthodes d'accès, comme mentionné précédemment, et peuvent également inclure une fonction d'alarme.

Dans le prochain chapitre, nous parlerons de la serrure électronique que nous allons développer. Nous expliquerons son fonctionnement et comment elle peut gérer les tentatives de piratage.

## **Chapitre II : Présentation des composants du système étudié**

## Chapitre II : Présentation des composants du système étudié

### 1. Introduction

L'évolution constante des technologies de l'information et de la communication a ouvert des perspectives novatrices dans le domaine de la sécurité résidentielle et commerciale. Parmi ces avancées, la conception d'une serrure électronique intelligente, combinant un système d'accès par code PIN, la technologie d'identification par radiofréquence (RFID) et une interface de contrôle via une application mobile, représente une solution prometteuse pour renforcer l'accès et la gestion des ouvertures. Ce chapitre explorera en détail les différentes étapes de la conception d'un tel système, en se concentrant sur la sélection des composants matériels essentiels.

### 2. Unité de commande et de traitement

#### 2.1. ESP32

Le module ESP-32S est un système sur puce (SoC) à faible coût et faible consommation d'énergie, intégrant Wi-Fi et Bluetooth bi-mode. Ce module est basé sur le microprocesseur dual-core 32 bits Xtensa® LX6, offrant une grande puissance de traitement et de connectivité pour les projets IoT.

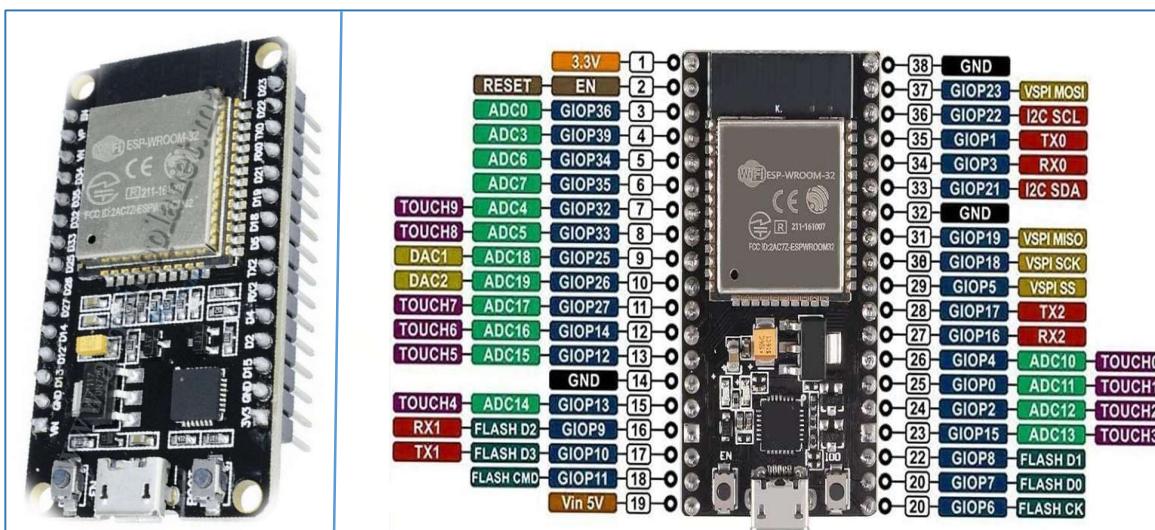


Figure 11 : ESP32 WROOM 32

Figure 10 : ESP32 WROOM 32 GPIO Pins

Fonctionnement : Premièrement, l'ESP32-WROOM-32S est alimenté en 3.3V via les broches d'alimentation de la carte sur laquelle il est monté ou via son connecteur micro-USB intégré sur certaines cartes de développement l'utilisant. Cette alimentation est le point de départ de toutes

ses opérations. Deuxièmement, la programmation de l'ESP32-WROOM-32S se fait typiquement en téléchargeant un programme (firmware) compilé en C/C++ à travers sa connexion USB-série (intégrée sur la plupart des cartes de développement) en utilisant l'ESP-IDF ou l'IDE Arduino avec le support ESP32 installé. On peut également utiliser d'autres outils et langages compatibles. Troisièmement, au démarrage, le bootloader préprogrammé dans la ROM du module s'active. Il initialise les composants essentiels et recherche le firmware dans la mémoire flash SPI intégrée au module ESP32-WROOM-32S. Des broches de “strapping” sur le module peuvent influencer le mode de démarrage. Quatrièmement, une fois le firmware chargé et exécuté par les deux cœurs LX6 du module ESP32-WROOM-32S, le code peut interagir avec les nombreuses broches GPIO disponibles sur le module pour contrôler des périphériques externes, lire des capteurs via les ADC intégrés, générer des signaux analogiques avec les DAC, ou communiquer via les interfaces SPI, I2C et UART également accessibles via ses broches. Cinquièmement, l'ESP32-WROOM-32S excelle dans la connectivité sans fil grâce à sa puce ESP32 intégrée, offrant une connexion Wi-Fi (802.11 b/g/n) pour se connecter à des réseaux locaux ou à internet, et le Bluetooth (Classic et Low Energy) pour communiquer avec d'autres appareils à courte portée. Ces fonctionnalités sont cruciales pour les applications IoT. Enfin, pour les applications alimentées par batterie, l'ESP32-WROOM-32S supporte différents modes de gestion de l'énergie, permettant de réduire sa consommation électrique lorsqu'il est inactif, prolongeant ainsi l'autonomie du système. [16]

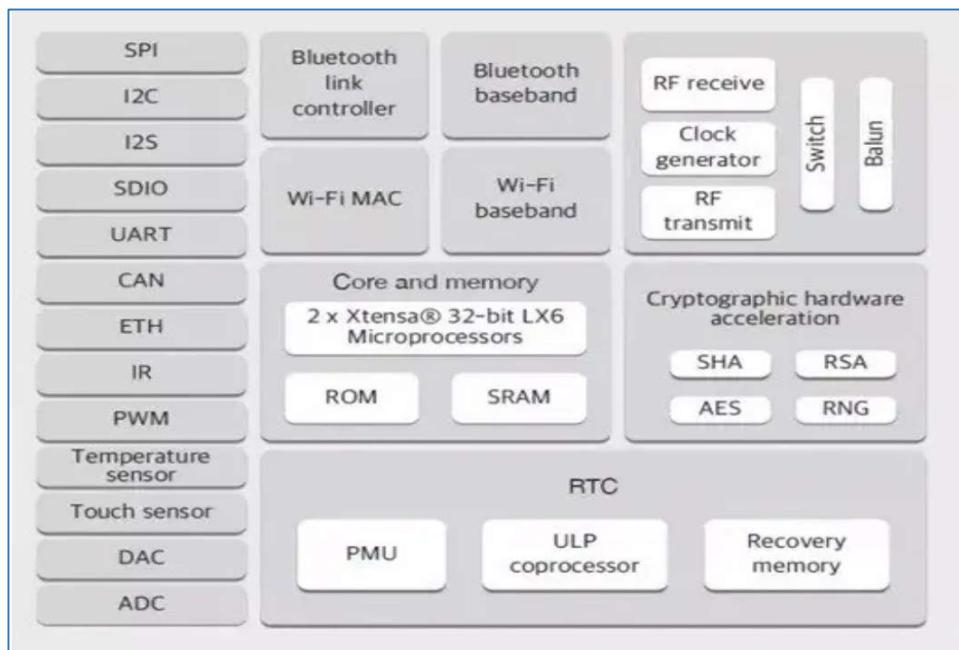


Figure 12 : Schéma simplifié de la carte ESP32-WROOM-32S.

## Caractéristiques de la carte ESP32-WROOM-32S

- Microcontrôleur: Xtensa® Dual-Core 32-bit LX6
- Vitesse d'horloge: Jusqu'à 600 DMIPS
- Convertisseur USB: CP2102
- Connecteur USB: Micro USB
- Mémoire Flash: Jusqu'à 64 Mbytes QSPI Flash / SRAM
- Mémoire RAM: 536 Kbytes
- E/S Numériques: Plus de 26 broches GPIO utilisables
- PWM: 19 canaux PWM (16 canaux 16 bits + 3 canaux 16 bits pour moteurs)
- Sorties Analogiques: 2 canaux 8 bits
- Entrées Analogiques: 16 canaux 12 bits
- Communications: 3 Série (2 UART + 1 UART avec contrôle de flux), 3 SPI, 2 I2C, 2 I2S
- Capteurs Intégrés: 8 canaux IR, Capteurs tactiles via GPIO
- Bluetooth: Bluetooth v4.2 BR / EDR et BLE
- WiFi: Intégré 802.11 b/g/n 2.4GHz
- Boutons: EN (Reset), BOOT
- Programmation: Compatible avec Arduino IDE (via l'ajout de bibliothèques ESP32), Lua, ESP-IDF
- Espacement des Broches: Espacement de 0.1", espacement entre les rangées d'environ 0.9", Compatible avec les plaques d'essai
- Tension de Fonctionnement: 2.2V – 3.6V (pour l'alimentation), 3.3V pour les GPIO

## 3. Unité communication

### 3.1. Péphérique d'entrée

#### 3.1.1. RFID (Identification par Radiofréquence)

RFID est une technologie qui permet l'identification et le suivi sans contact d'objets à l'aide d'ondes radio .Les systèmes RFID se composent de deux composants principaux : les tags et les lecteurs. Les tags (badges) sont de petits appareils qui stockent et transmettent des informations à distance à l'aide d'ondes radio. Les lecteurs, quant à eux, sont responsables de la réception des informations transmises par ces tags. [17]



Figure 13 : module RC522 RFID

### 3.1.1.1. Lecteur de carte RFID

Le RC522 est basé sur la puce MFRC522, qui fonctionne à une fréquence de 13,56 MHz. Il est capable de lire et d'écrire des données sur des tags RFID compatibles, y compris les cartes

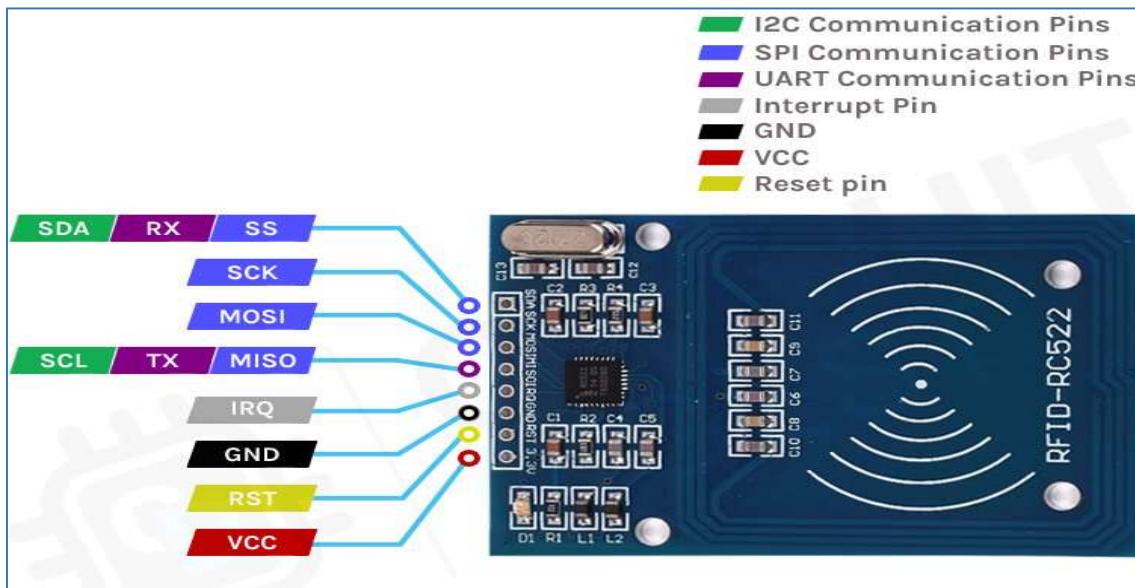


Figure 14 : schéma d'un RFID-RC522

MIFARE (un type de carte à puce sans contact utilisée à des fins très diverses, notamment le contrôle d'accès, les transports publics et les systèmes de paiement) ce qui en fait un choix idéal pour une large gamme d'applications telles que les systèmes de contrôle d'accès, la gestion des stocks et les projets interactifs [17]

**VCC** : Cette broche alimente le module. Vous pouvez connecter une tension d'alimentation comprise entre 2,5 et 3,6 V.

**RST** : Cette broche agit comme un circuit de mise hors tension programmable. Lorsqu'elle reçoit un signal bas, le courant est coupé et toutes les broches sont pratiquement déconnectées.

**GND** : Cette broche est la masse du module.

**IRQ** : Cette broche est la broche d'interruption. Les broches d'interruption alertent le microcontrôleur lorsqu'une balise est présente.

**MISO/SCL TX** : Cette broche peut être utilisée pour SPI MISO (Master IN Slave OUT), I2C SCL (horloge série) ou UART TX.

**MOSI** Cette broche peut être utilisée pour SPI MOSI (Master OUT Slave IN)

**SCK** : Cette broche est destinée à SPI SCK (horloge série).

**SDA/SS/RX** : Cette broche peut être utilisée pour I2C SDA (données série), SPI SS (entrée de signal) et UART RX. [18]

Caractéristique de RFID-RC522 :

- Courant de fonctionnement : 13-26 mA/DC 3,3 V
- Courant de repos : 10-13 mA/DC 3,3 V
- Courant de veille : <80 uA
- Courant de crête : <30 mA
- Fréquence de fonctionnement : 13.56 MHz
- Types de cartes pris en charge : mifare1 S50, mifare1 S70 MIFARE Ultralight, Mifare Pro, MIFARE DESFire
- Température ambiante de fonctionnement : -20-80 degrés Celsius
- Température ambiante de stockage : -40-85 degrés Celsius
- Humidité relative : humidité relative de 5 % à 95 %
- Distance de lecture :  $\geq 50$  mm/1,95' (Mifare 1)
- Taille du module :  $40 \times 60$  mm/1.57\*2.34'Paramètre des interfaces du module SPI
- Taux de transfert de données : 10 Mbit/s maximum

### 3.1.1.2. Tag (badge) RFID

Le badge MIFARE est un support d'identification électronique utilisé pour sécuriser les locaux et les données des entreprises. Il fonctionne avec un lecteur RFID pour lire les informations stockées sur la puce électronique, telles que des numéros de série, des clés de chiffrement ou d'autres données spécifiques selon les besoins des entreprises. La technologie

MIFARE est largement utilisée dans le monde entier pour assurer la sécurité des bâtiments, des machines, et des données confidentielles



Figure 15 : schéma d'un tag RFID

Fonctionnement d'un badge RFID :

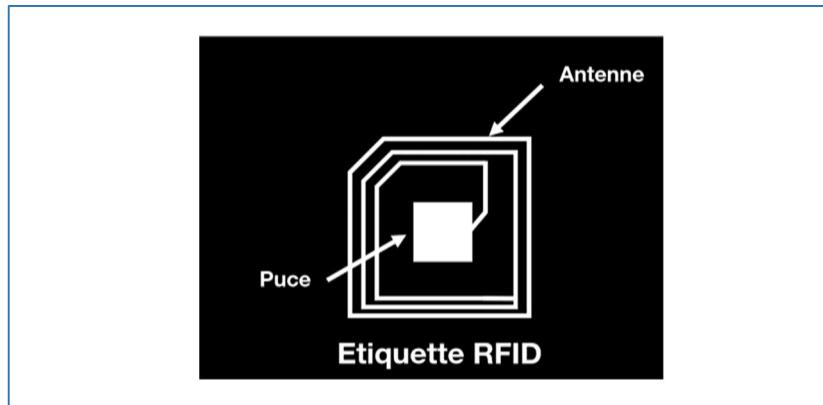


Figure 16 : schéma d'une étiquette RFID

Un badge RFID, pour Radio Frequency Identification, est un petit objet qui contient une puce électronique et une antenne. Il peut être lu à distance par un lecteur RFID grâce à une connexion sans fil. Le lecteur envoie un signal radio à la puce, qui répond en envoyant les informations stockées sur celle-ci. Les informations peuvent être stockées sur une carte magnétique, une clé USB ou même un bracelet. [19]

### 3.1.1.3. Fonctionnement de la communication de RFID

La communication entre votre badge RFID et le système de lecture est basée sur la technologie de communication sans fil. Le système de lecture émet des ondes électromagnétiques à une fréquence donnée, qui sont ensuite captées par l'antenne du badge

RFID. Le badge répond alors en renvoyant un signal radio contenant les informations stockées dans sa puce électronique.

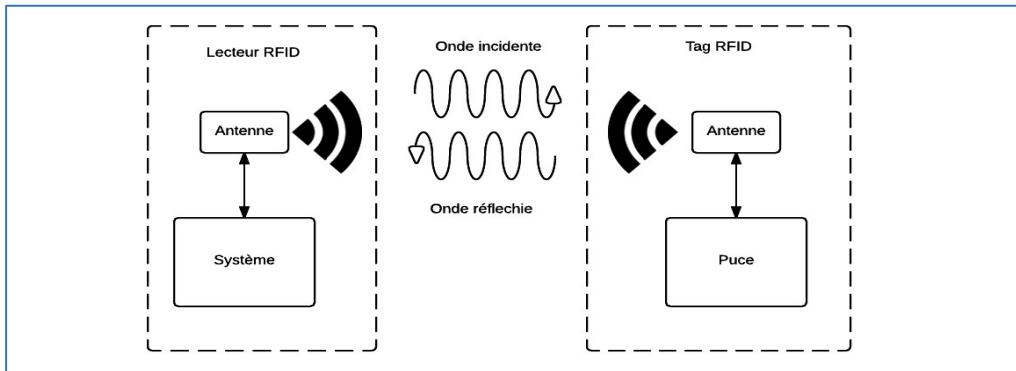


Figure 17 : schéma de fonctionnement RFID

Le système de lecture reçoit ensuite ce signal radio et extrait les informations contenues dans celui-ci. Les informations peuvent être un simple identifiant unique, comme un numéro de série ou une chaîne de caractères contenant des informations supplémentaires sur le badge. [20]

### 3.1.2. Clavier matriciel 4x4

Un clavier matriciel  $4 \times 4$  est un type de clavier qui utilise une grille de 4 colonnes et 4 lignes de touches pour entrer des données. Chaque touche est associée à une combinaison unique de lignes et de colonnes, ce qui permet de détecter la touche appuyée en mesurant la résistance entre les lignes et les colonnes. [21]

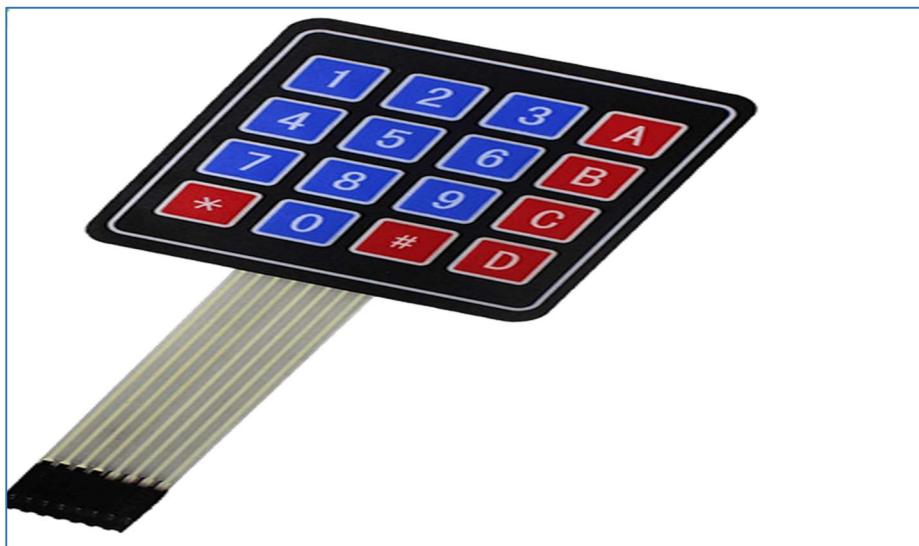


Figure 18 : clavier matriciel

Fonctionnement d'un clavier matriciel : Ils existent plusieurs claviers selon le nombre de boutons qu'ils contiennent, ils peuvent se configurer en  $3 \times 3$ ,  $3 \times 4$ ,  $4 \times 4$ ...

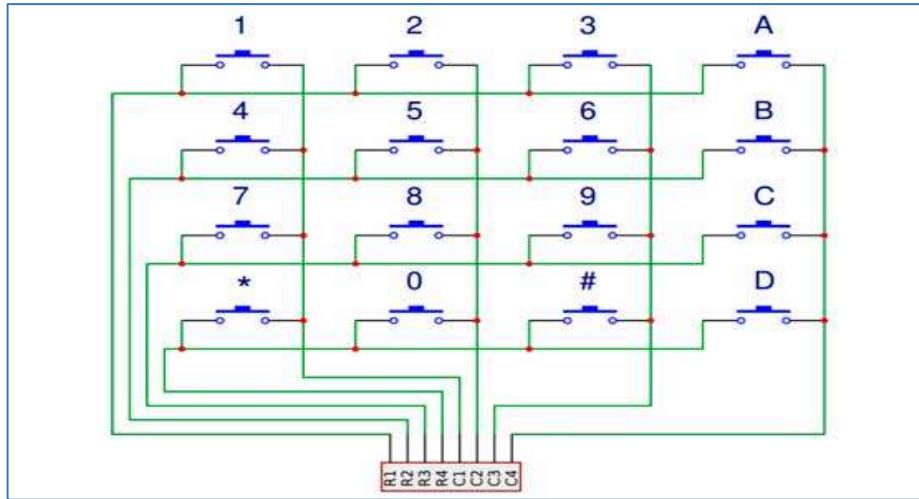


Figure 19 : schéma de connexion d'un clavier matriciel

Ils sont constitués de trois membranes superposées; celle du milieu, non conductrice, sépare deux autres membranes fabriquées en matériaux conducteurs.

L'interrupteur est toujours ouvert et, en pressant un bouton, la membrane supérieure et l'inférieure prennent contact laissant passer le courant. [21]

### 3.2. Pérophérique de sortie

#### 3.2.1. Afficheur LCD I2C

Un afficheur I2C LCD est un type d'afficheur à cristaux liquides (LCD) qui utilise le protocole de communication en série I2C (Inter-Integrated Circuit) pour se connecter à un microcontrôleur ou à un autre dispositif de traitement de données.

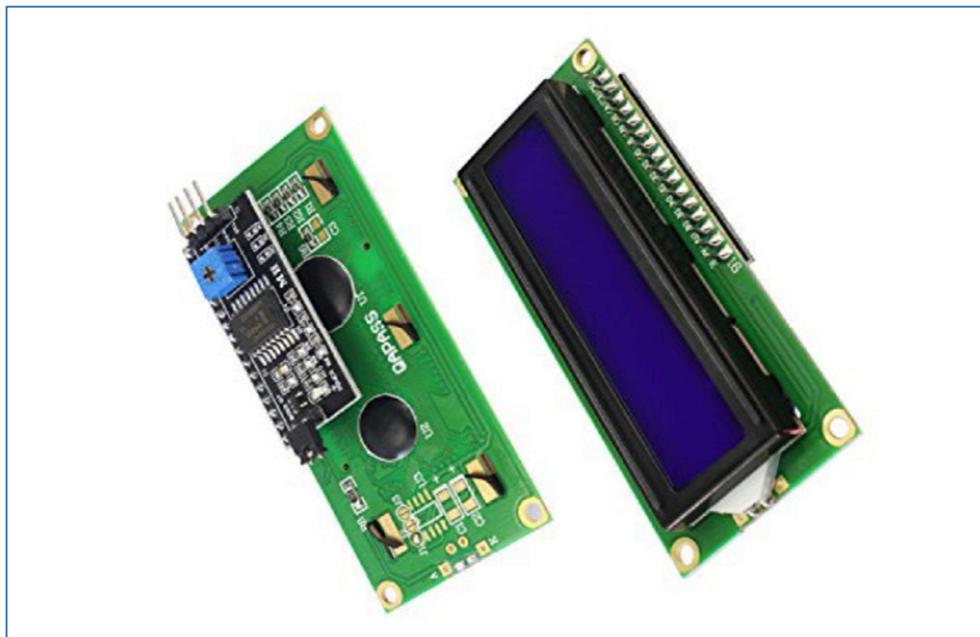


Figure 20 : LCD I2C

Les afficheurs I2C LCD sont couramment utilisés dans les projets électroniques pour afficher des données et des informations, comme des messages, des valeurs de capteurs et d'autres données. Ils sont souvent utilisés dans les projets de robotique et de domotique, ainsi que dans les appareils électroniques portables et les dispositifs de contrôle industriels. [22]

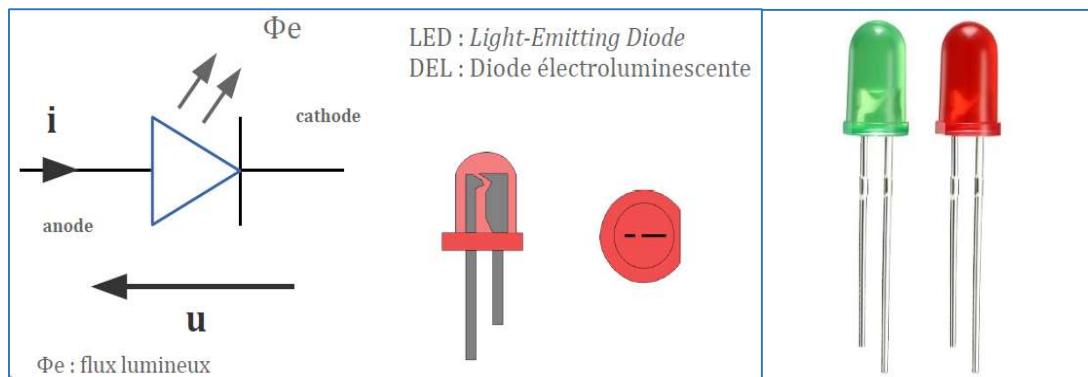
Pour utiliser un afficheur I2C LCD, il est nécessaire de le connecter à un microcontrôleur ou à un autre dispositif de traitement de données en utilisant les broches de communication SDA (données) et SCL (horloge). En utilisant des bibliothèques de logiciels ou des instructions de programmation spécifiques, il est possible d'envoyer des données à l'afficheur et de les afficher sur l'écran. Les afficheurs I2C LCD sont connus pour leur faible consommation d'énergie et leur simplicité de configuration et d'utilisation. [23]

Spécifications techniques de LCD I2C :

- Alimentation : 5V
- 2 lignes de 16 caractères
- Rétro-éclairage vert/bleu, caractères blancs
- Adresse I2C : 0x20
- Potentiomètre pour réglage du contraste
- Dimensions : 36 x 79 x 20 mm
- Poids : 40g

### 3.2.2. Diode électroluminescente LED

La LED (Light Emitting Diode) est textuellement une « diode émettant de la lumière ». Elle est également appelée DEL (Diode électroluminescente) en français et SSL (Solid State Lighting) en anglais. Il s'agit d'un composant (opto) électronique qui, d'une part, ne laisse passer le courant électrique que dans un sens (définition de la diode) et d'autre part, émet de la lumière.



Les LED fonctionnent suivant le principe de la luminescence et plus exactement de l'électroluminescence, puisque la lumière est émise suite au passage d'un courant électrique. Le principe est de produire un déficit d'électron dans une zone (matériau dopé p) et un excédent dans une autre (matériau dopé n) : à la jonction entre les deux matériaux, les « trous » d'électrons se recombinent avec les électrons et génèrent un photon. [24]

### 3.2.3. Buzzer électronique

Un buzzer est un composant électronique capable de convertir l'énergie électrique en vibrations mécaniques, produisant du son, grâce à des matériaux piézoélectriques qui ont la capacité de vibrer lorsqu'ils sont exposés à un champ électrique.

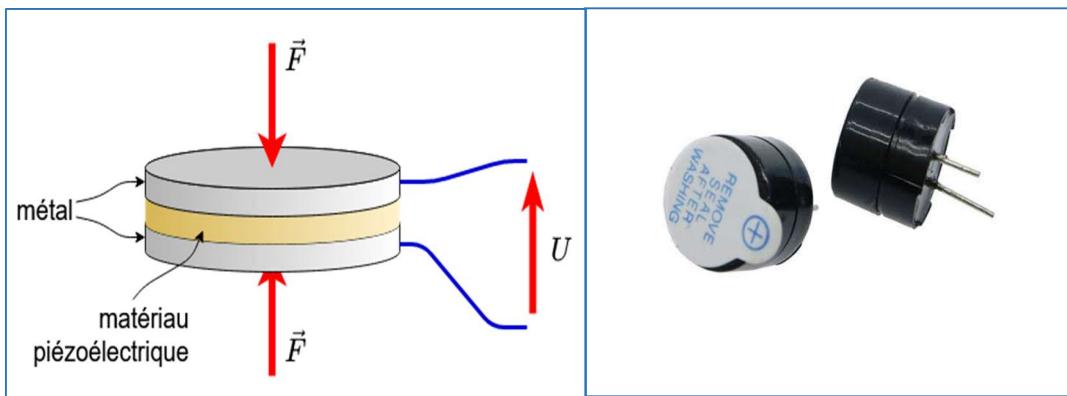


Figure 23 : schéma de fonctionnement de buzzer

Figure 24 : Buzzer électronique

Le principe de fonctionnement de la cloche piézoélectrique est simple mais efficace. Il est basé sur l'effet piézoélectrique, un phénomène physique où certains matériaux produisent des charges électriques lorsqu'ils sont soumis à une contrainte mécanique. À l'inverse, ces mêmes matériaux se déforment lorsqu'ils sont exposés à une tension électrique. Dans le cas de la cloche, c'est ce dernier côté qui est utilisé. Lorsqu'un courant électrique traverse le matériau piézoélectrique, celui-ci vibre, générant une onde sonore. Ces vibrations sont ensuite amplifiées pour produire un son suffisamment fort pour être entendu.

La majorité des buzzers piézoélectriques sont conçus pour fonctionner à une fréquence de résonance spécifique, généralement comprise entre 2 et 4 kHz, ce qui correspond à la plage idéale pour les alarmes audibles. [25]

Spécifications techniques du buzzer :

- Type : Buzzer actif
- Tension nominale : 5V DC
- Plage de tension de fonctionnement : 4V à 7V DC

- Courant nominal : <30 mA
- Type de son : Bip continu
- Fréquence de résonance : ~2300 Hz
- Niveau sonore : > 85 dB à 10 cm
- Dimensions : 19mm x 15mm. [26]

## 4. Unité des Actionneurs

### 4.1. Serrure électromagnétique

La serrure électromagnétique solénoïde 12 V est une option électronique efficace et sécurisée, largement utilisée dans des applications telles que les coffres-forts et les portes de maison.

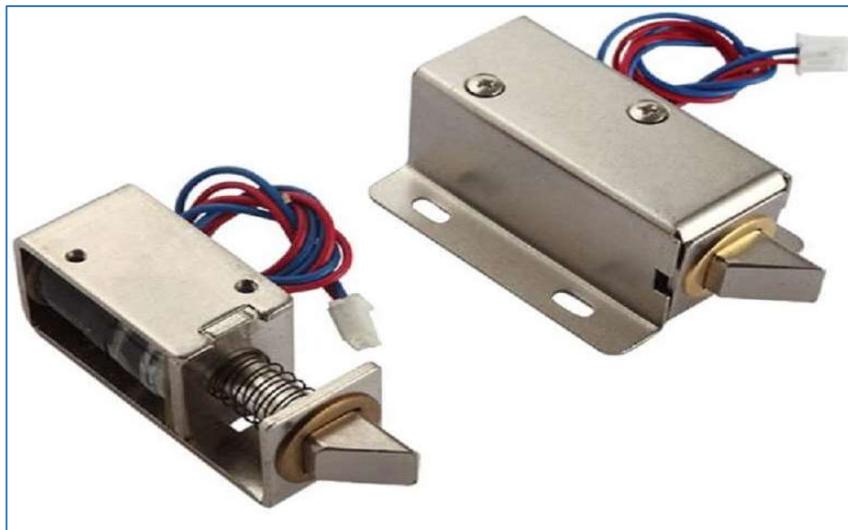


Figure 25 : Serrure électromagnétique

Cette serrure repose sur un électroaimant qui fonctionne lorsqu'il est connecté à 9 à 12 volts CC, maintenant la porte fermée. Lorsque la tension est activée, la serrure est immédiatement déverrouillée, ce qui permet d'ouvrir la porte. Ce qui est génial, c'est qu'il ne consomme pas d'énergie lorsqu'il est éteint, ce qui le rend extrêmement économique en énergie. [27]

Les Avantages :

- Installation facile : Sa conception simple avec patte coudée et support de montage bien pensé rend l'installation rapide et permet de gagner du temps dans la mise en place de projets de verrouillage électronique.
- Efficacité énergétique : Il ne consomme aucune énergie à l'arrêt, ce qui contribue à améliorer l'efficacité du système à long terme.

- Vitesse d'ouverture élevée : Il ne faut qu'une seconde pour l'ouvrir, ce qui le rend idéal dans les endroits qui nécessitent sécurité et réponse rapide.
- Conception robuste : Durable et fiable, avec un montage sécurisé et un câblage sécurisé garantissant des performances durables.
- Intégration facile : Grâce à sa petite taille, il vous permet de l'installer facilement.

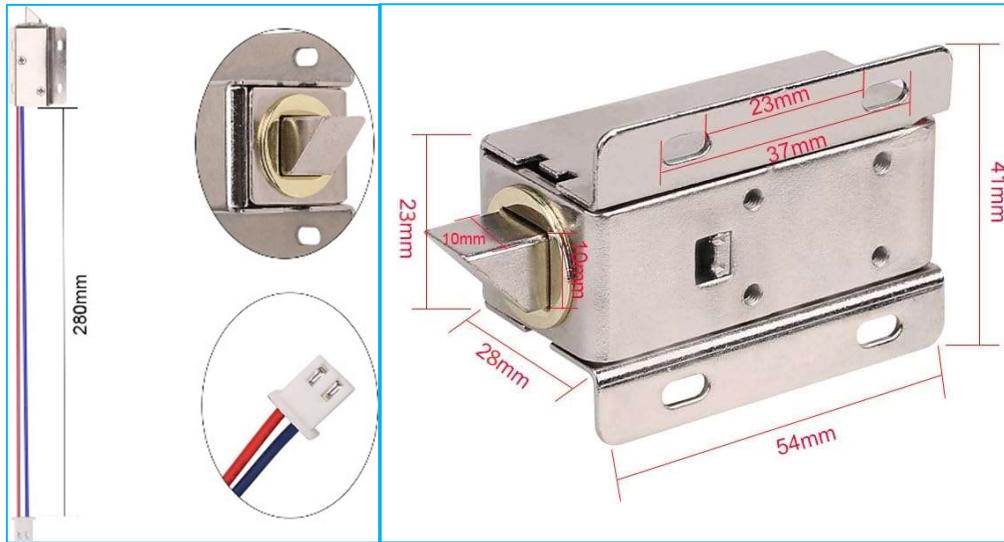


Figure 26 : Caractéristiques Techniques de serrure électromagnétique

Caractéristiques Techniques :

- Tension de fonctionnement : 12VDC
- Temps de déverrouillage : 1 seconde
- Courant d'entrée : 0,4 A
- Dimensions: longueur 54 mm ; Largeur 23 mm ; Hauteur 28 mm ; Longueur de la langue 7 mm ; Hauteur de la langue – 10 mm
- Longueur du câble : 200-280 mm

## 4.2. Module Relais 5v

Le module relais 5V à un canal est un dispositif de commutation utilisé pour contrôler des charges AC et DC. Ce module est compatible avec les microcontrôleurs tels qu'Arduino et esp32. Permettant de contrôler des appareils électriques à distance.

Le relais fonctionne en utilisant une bobine électromagnétique activée par une tension de commande de 5V pour fermer ou ouvrir les contacts. Cela permet de contrôler des charges

électriques élevées avec une faible tension de commande, garantissant une isolation efficace et une commutation fiable. [28]



Figure 27 : Relais

Caractéristiques :

- Type de relais: Relais électromécanique
- Nombre de canaux: 1
- Tension de commande: 5V DC
- Courant nominal: 10A
- Configuration des contacts: SPDT (Single Pole Double Throw)
- Tension de charge: AC250V 10A, DC30V 10A
- Indicateurs LED: Indique l'état du relais
- Dimensions: 92 x 46 x 20 mm
- Déclenchement: LOW Level

## 5. Application mobile

### 5.1. Application Blynk

Blynk est une plateforme pour l'Internet des Objets (IoT). Elle permet notamment de Concevoir une application mobile (Android et iOS) pour contrôler et visualiser les données d'un système embarqué via un serveur Cloud public ou privé. [29]

La conception de l'application mobile (Android et Ios) à base de widgets (éléments Graphiques) est réalisée par simple glisser & déposer sans écrire une ligne de code.



Figure 28 : Application Blynk

Blynk a été conçu pour l'Internet des objets. C'est une plate-forme avec des applications Ios et Android pour contrôler Arduino, Raspberry Pi, ESP32 et autres sur Internet. Il peut également :

- Contrôler le matériel à distance,
- Afficher les données des capteurs,
- Stocker des données, les visualiser et faire beaucoup d'autres choses intéressantes.

La plate-forme comprend trois (03) composants principaux :

- Blynk App: nous permet de créer des interfaces incroyables pour notre projet en utilisant les divers widgets que nous fournissons.
- Blynk server: responsable de toutes les communications entre le smartphone et le matériel. Nous pouvons utiliser notre Blynk Cloud ou exécuter notre serveur Blynk privé localement. Il est open-source, pourrait facilement gérer des milliers d'appareils et peut même être lancé sur un Raspberry Pi.
- Les bibliothèques Blynk : pour toutes les plates-formes matérielles populaires permettent la communication avec le serveur et traitent toutes les commandes entrantes et sortantes.

Fonctionnalités :

- API et UI similaire pour tous les hardwares et périphériques supportés
- Connexion au cloud via: Ethernet – Wi-Fi - Bluetooth et BLE - USB (Serial)
- Collection de widgets faciles à utiliser
- Manipulation des broches directes sans code à écrire
- Facilité d'intégrer et ajouter de nouvelles fonctionnalités en utilisant les broches virtuelles
- Surveillance de l'historique des données via le widget History Graph
- Communication Périphérique-à-Périphérique en utilisant le widget Bridge
- Envoi d'emails, de tweets, de notifications push, etc.

## 5.2. Communication entre ESP32 et application mobile

Le fonctionnement de la connexion entre la carte ESP32 et la plateforme Blynk repose sur l'établissement d'une liaison sans fil via un réseau Wi-Fi local. Dans un premier temps, la carte se connecte à Internet en utilisant le réseau Wi-Fi disponible, pour lequel le nom du réseau (SSID) et le mot de passe sont programmés dans le code. Une fois la connexion Internet établie avec succès, la carte commence à communiquer avec les serveurs cloud de la plateforme Blynk en utilisant un jeton d'authentification unique (Auth Token) obtenu lors de la création d'un nouveau projet dans l'application.

Cette plateforme sert de pont de communication entre l'application smartphone et la carte ESP32. Elle permet de transmettre les commandes et instructions depuis l'application vers la carte, tout en transférant simultanément les données et les lectures des capteurs de la carte vers l'application. La connexion est maintenue de manière continue et sans interruption pendant toute la durée de fonctionnement du système. Cela permet à l'utilisateur de contrôler la carte et de surveiller les données en temps réel depuis n'importe quel endroit, à condition que la carte et le smartphone soient tous deux connectés à Internet.

## 6. Conclusion

En conclusion, ce chapitre a posé les bases de la conception de notre serrure électronique intelligente en détaillant les composants matériels essentiels. Nous avons exploré l'unité de commande et de traitement ESP32, les périphériques d'entrée pour la saisie d'un code alphanumérique (clavier matriciel) et l'identification RFID (lecteur et carte), ainsi que les périphériques de sortie pour l'affichage (LCD) et la signalisation sonore (buzzer et LED). De plus, nous avons présenté l'unité des actionneurs, constituée d'une serrure électromagnétique responsable du verrouillage, et l'application mobile, sans oublier les méthodes de communication WIFI et Bluetooth nécessaires à l'interaction et au contrôle à distance du système.

# Chapitre III : Simulation et Réalisation du Système

## Chapitre III : Simulation et Réalisation du Système

### 1. Introduction

Ce chapitre présent les différentes étapes de la conception de notre projet, ainsi que le fonctionnement de chaque élément, et après avoir étudié les différents composants que nous utilisons pour notre projet en classe précédemment, la réalisation de système fait l'objet de ce dernier chapitre qui illustre deux parties, la réalisation matérielle avec définition de toutes les étapes nécessaires pour réaliser ,et enfin la réalisation logiciel qui concerne la programmation de la carte Esp32 pour la mise en marche de système.

### 2. Simulation du système

#### 2.1. Programme wokwi

WOKWI est un simulateur électronique gratuit et open source, utilisé pour simuler des circuits et des projets électroniques de manière virtuelle, sans avoir besoin de composants réels. Il se distingue par sa compatibilité avec plusieurs plateformes populaires telles qu'Arduino, ESP32 et Raspberry Pi Pico. Il prend également en charge plusieurs langages de programmation comme Arduino C et Micro Python, ce qui le rend adapté à différents types d'utilisateurs, qu'ils soient débutants ou expérimentés. WOKWI fonctionne directement depuis un navigateur, sans nécessiter l'installation de logiciels, ce qui le rend facile d'accès et simple à utiliser. Cet outil est très utile dans le domaine de l'éducation et de l'apprentissage autonome, car il permet aux utilisateurs de tester et de valider leurs projets virtuellement, ce qui aide à réduire les erreurs et à gagner du temps et des efforts avant de passer à la réalisation pratique.

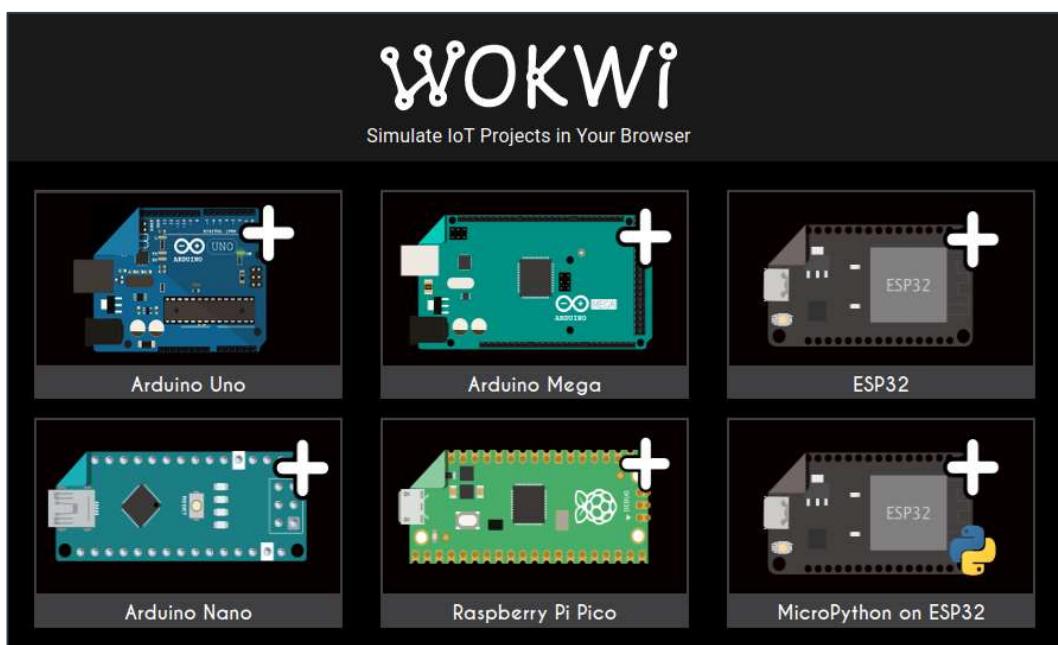


Figure 29 : programme wokwi

## 2.2. Câblage des composants à l'esp32

### 2.2.1. Alimentation d'esp32

Esp32 peut être alimentée par une connexion USB ou une source d'alimentation externe comme une batterie.

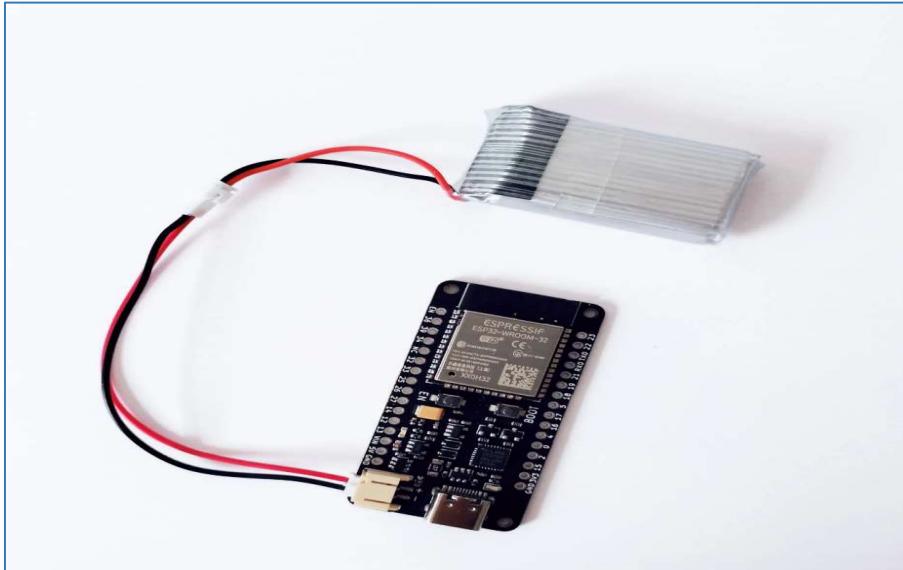


Figure 30 : Alimentation d'esp32

### 2.2.2. Clavier matriciel 4x4

On connecte les 8 sorties du clavier aux 8 broches de la carte ESP32 suivant cet ordre:

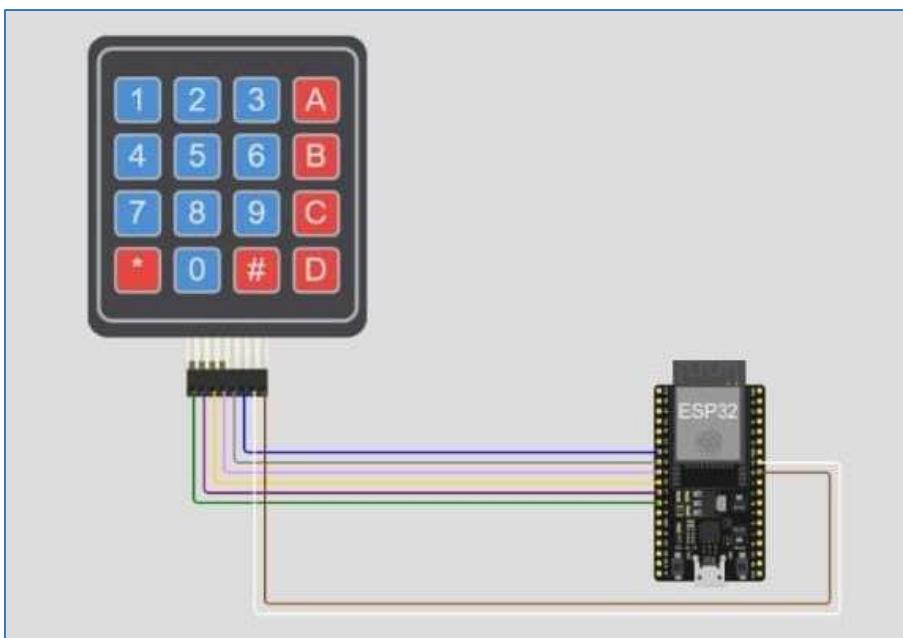


Figure 31 : câblage du keypad

- la broche R1 à la broche 14 de la carte ESP32
- la broche R2 à la broche 27 de la carte ESP32
- la broche R3 à la broche 26 de la carte ESP32
- la broche R4 à la broche 25 de la carte ESP32
- la broche C1 à la broche 33 de la carte ESP32
- la broche C2 à la broche 32 de la carte ESP32
- la broche C3 à la broche 19 de la carte ESP32
- la broche C4 à la broche 18 de la carte ESP32

### 2.2.3. LCD I2C

L'écran LCD est l'élément de visualisation de l'information entre l'utilisateur et le microcontrôleur, ça permet aussi d'afficher des messages en cas de fausse opération comme : "entrer le code", si c'est faux et < Bonjour> si c'est correct. Nous avons utilisé le LCD 16x2 qui contient 2 Ligne et 16 colons.

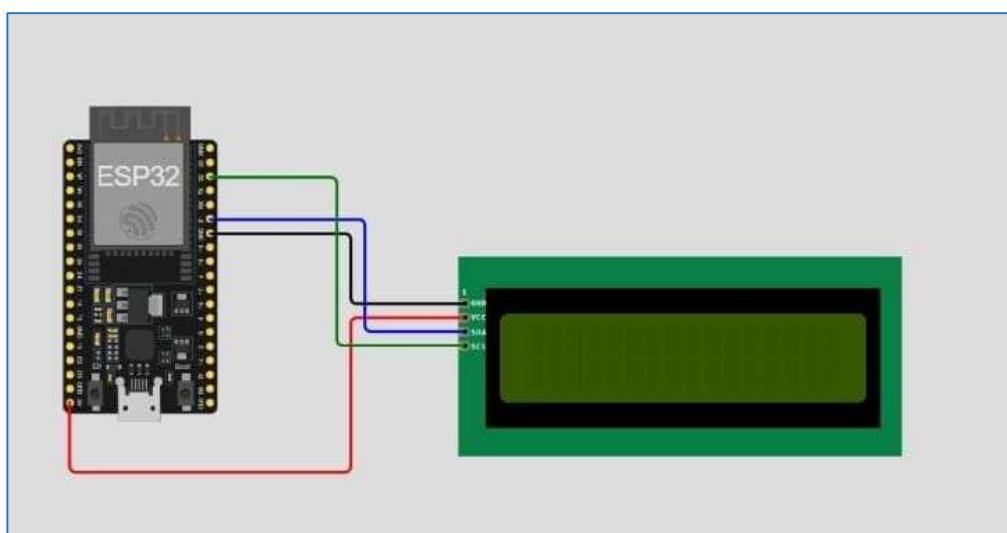


Figure 32 : câblage de LCD I2C

- connecter la broche VCC de l'afficheur à 5V de la carte ESP32.
- connecter la broche GND de l'afficheur à GND de la carte ESP32.
- connecter la broche SDA de l'afficheur à la broche 21 de la carte ESP32.
- connecter la broche SCL de l'afficheur à la broche 22 de la carte ESP32.

### 2.2.4. RFID RC522

- la broche SDA à la broche 5V de la carte ESP32
- la broche SCK à la broche 16 de la carte ESP32

- la broche MOSI à la broche 17 de la carte ESP32
- la broche MISO à la broche 2 de la carte ESP32
- la broche RST à la broche 15 de la carte ESP32
- la broche 3,3V à la broche 3,3V de la carte ESP32
- la broche GND à la broche GND de la carte ESP32

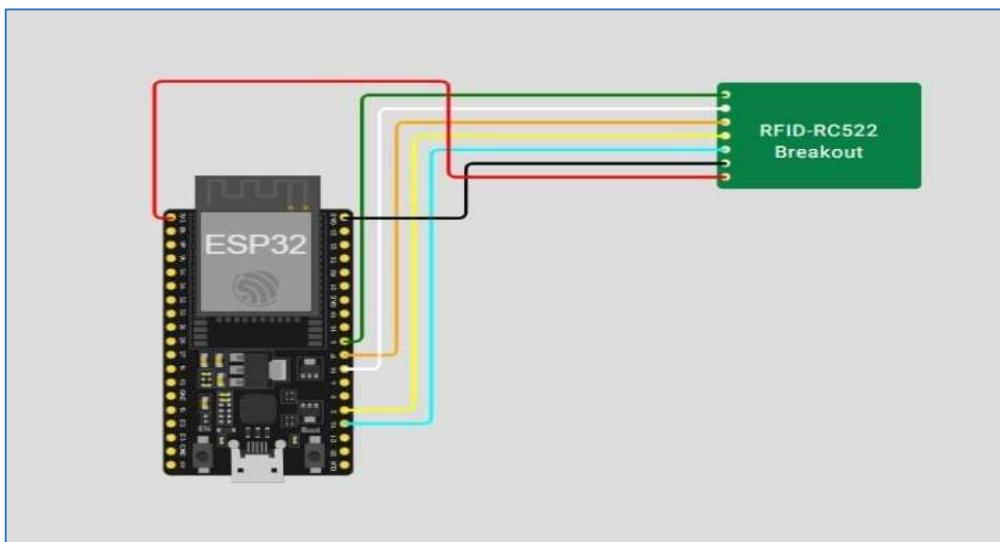


Figure 33 : câblage de RFID

### 2.2.5. Câblage des LED

On connecte la broche 4 de la carte ESP32 à la patte de la résistance de led vert et 12 de esp 32 a la résistance de led rouge et 0 de esp 32 a la résistance de led bleu.

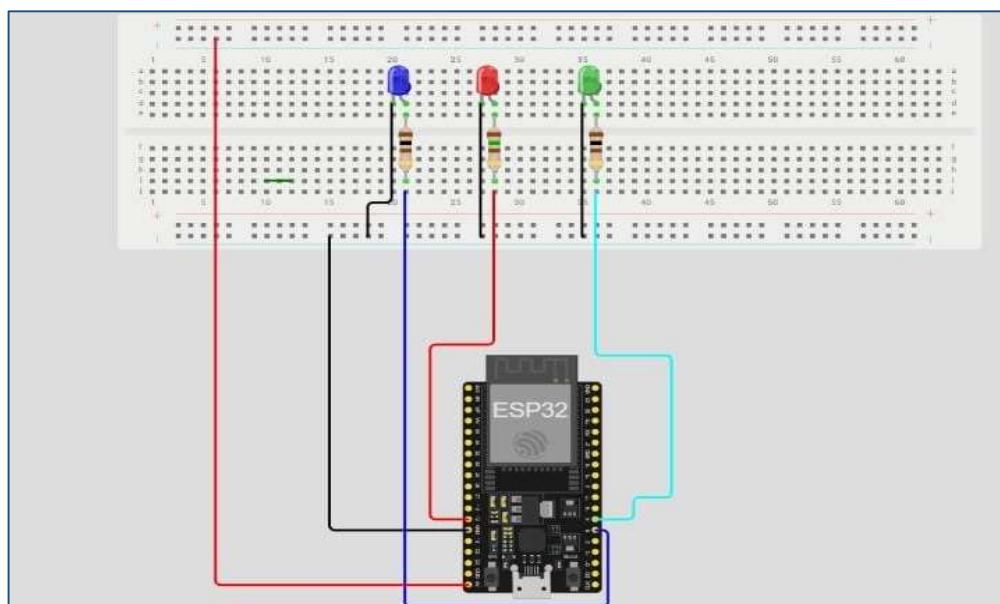


Figure 34 : câblage des LED

## 2.2.6. Buzzer

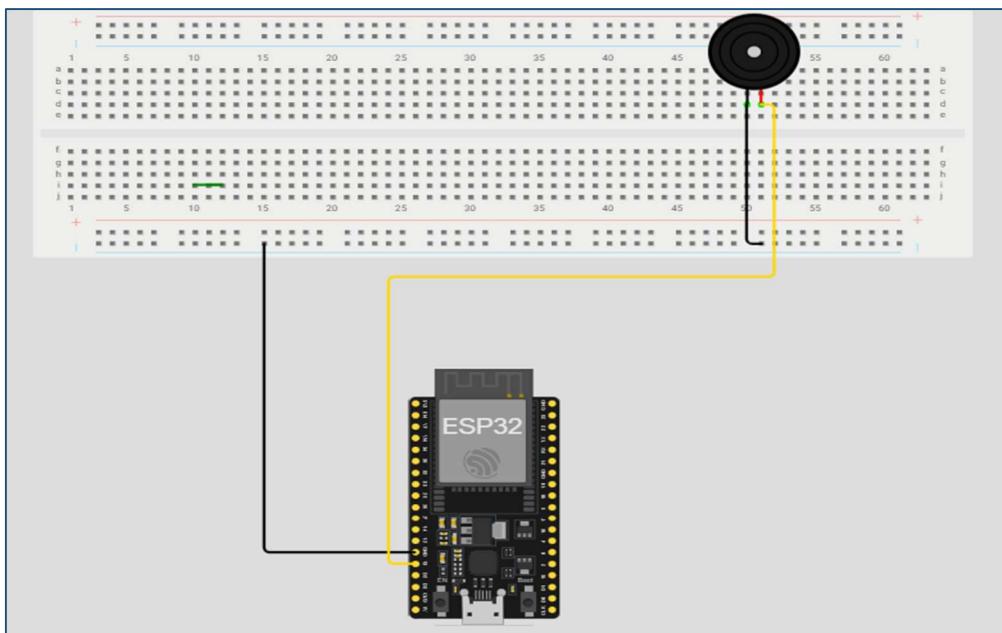


Figure 35 : connexion le buzzer avec l'esp32

- Branche (+) avec le pin 13 de l'esp32
- la masse (-) vers la GND

## 2.2.7. Relais

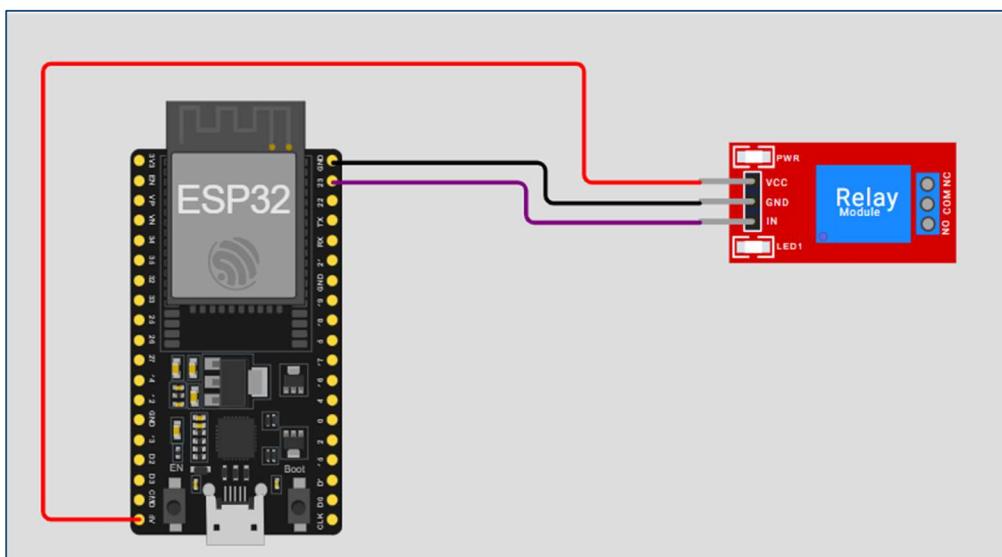


Figure 36 : connexion le relais avec l'esp32

- Branche in avec le pin 23 de l'esp32
- Branche le VCC avec le VCC de l'esp32
- Branche le GND avec le GND

### 2.2.8. Connexion d'un Serrure électromagnétique à l'esp32

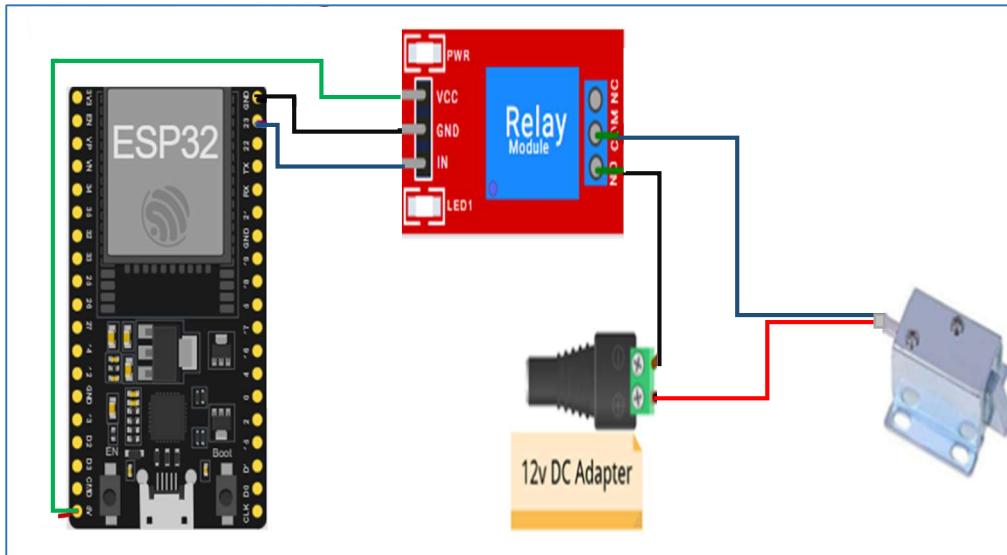


Figure 37 : connexion de la serrure avec l'esp32

- Branche la serrure avec le moins de l'adaptateur et avec le COM du relais
- Branche-le plus d'adaptateur avec le NO du relais

### 2.2.9. Schéma complet du système

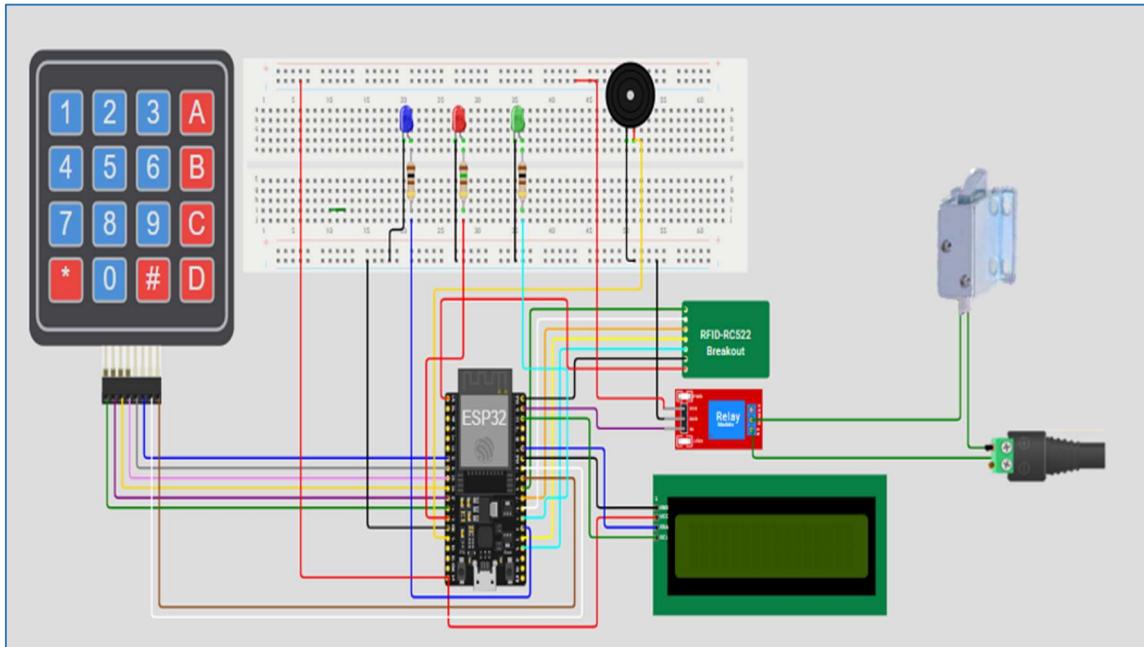


Figure 38 : schéma complet de système

### 3. Organigramme du système

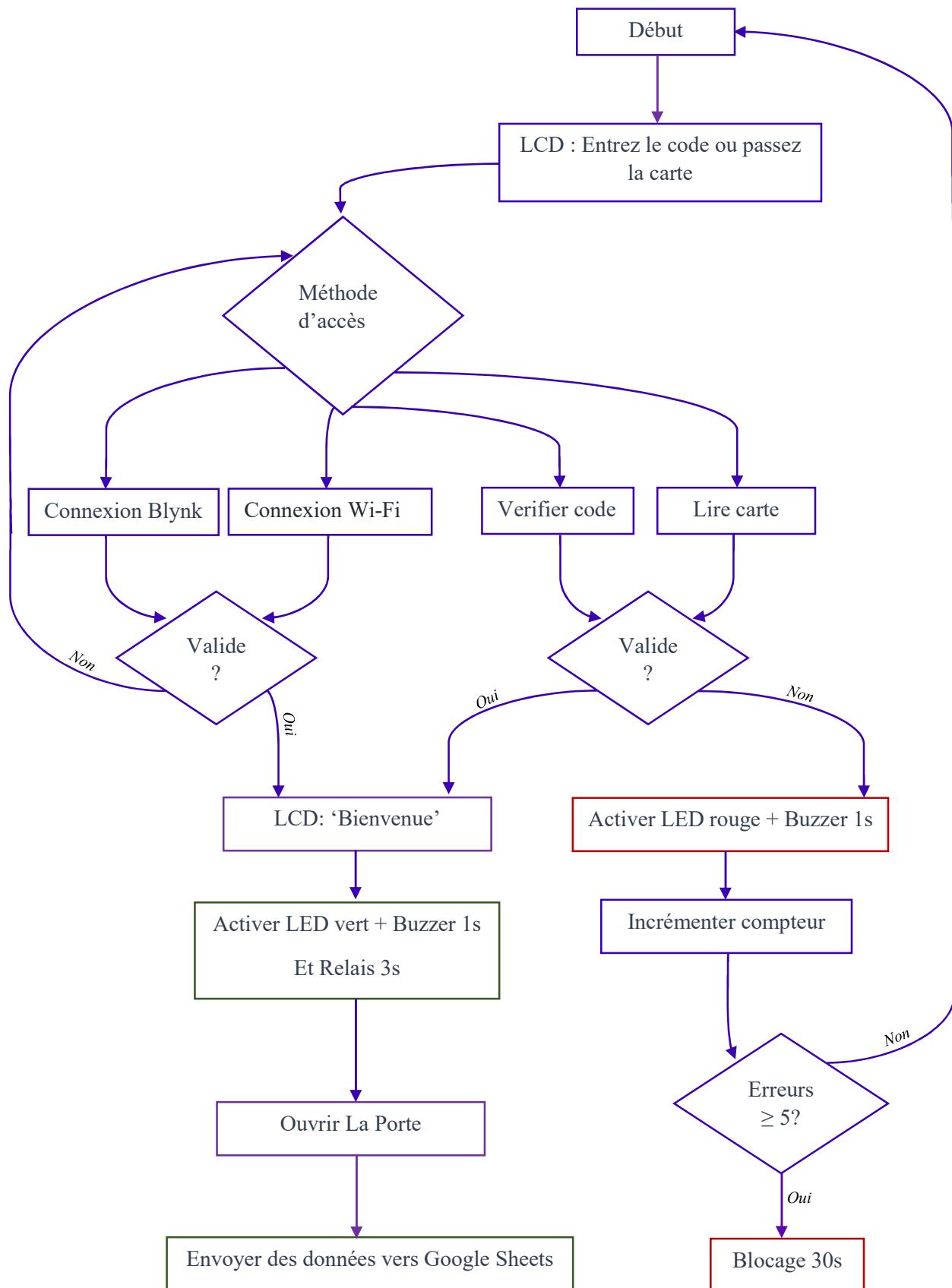


Figure 39 : Organigramme représentant le fonctionnement du système

## 4. Réalisation du système

### 4.1. Réalisation logicielle

#### 4.1.1. Arduino IDE

L'Arduino IDE (Integrated Development Environment) est un logiciel gratuit qui permet d'écrire, de compiler et de téléverser des programmes (croquis) sur les cartes Arduino. Les programmes créés pour Arduino sont appelés des croquis (sketches en anglais). Ce terme est hérité du logiciel Processing, qui utilisait déjà cette appellation pour encourager une approche créative et rapide de la programmation.

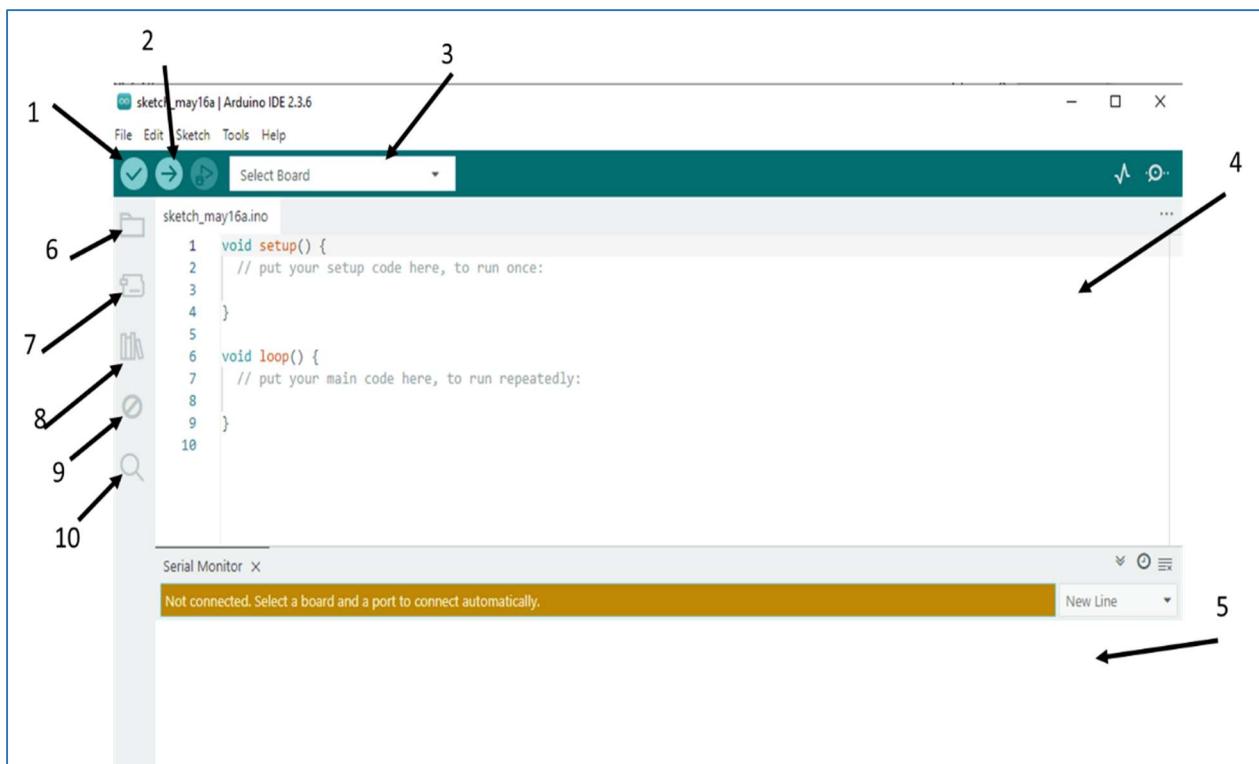


Figure 40 : la fenêtre d'Arduino IDE

L'interface dans le contexte de l'IDE Arduino est la partie visuelle et fonctionnelle du logiciel qui permet aux utilisateurs de programmer et de contrôler leur carte Arduino. Elle est composée de plusieurs éléments:

Barre de menus : Contient des menus déroulants pour accéder aux outils, paramètres et informations du programme:

1 Bouton Vérifier : Compile le programme pour vérifier la syntaxe et le transformer en une représentation adaptée à la carte Arduino.

2 Bouton Upload : Compile et télécharge le programme vers l'ESP32 connectée.

3 Select Board : Permet de choisir le modèle de carte que tu utilises (ex : Arduino Uno, ESP32...)  
Indispensable Avant toute vérification ou téléversement.

4 Fenêtre de l'éditeur : Zone où le code est écrit.

5 Moniteur Série : Permet de visualiser les données envoyées par Arduino et d'envoyer des données en retour.

6 Nouvelle onglet / Nouveau sketch : Crée un nouveau fichier. (Un nouveau projet). Par défaut, il s'appelle quelque chose comme sketch\_mai16a.ino.

Utilité : Démarrer un nouveau projet rapidement.

7 Ouvrir un sketch existant : Ouvre un sketch Arduino déjà enregistré sur ton ordinateur. Tu peux parcourir les dossiers pour trouver un ancien projet.

Utilité : Reprendre ou modifier un projet existant. Accéder à des exemples déjà faits.

8 Gérer les bibliothèques : Installer, mettre à jour ou supprimer des bibliothèques Arduino. Rechercher des bibliothèques par nom (par exemple : LiquidCrystal, Adafruit\_Sensor, WiFi.h, etc.).

Utilité : Ajouter des fonctionnalités supplémentaires à ton code (capteurs, affichage, WiFi, etc.).

9 Moniteur série : Ouvre ou ferme la fenêtre du moniteur série. Permet d'afficher les données envoyées depuis la carte (via Serial.print()).

Utilité : Déboguer le programme (voir les valeurs envoyées). Lire des messages, recevoir des données.

10 Rechercher dans le code Ouvre une barre de recherche dans ton code. Tu peux rechercher un mot, une fonction, une variable. Possibilité de remplacer également.

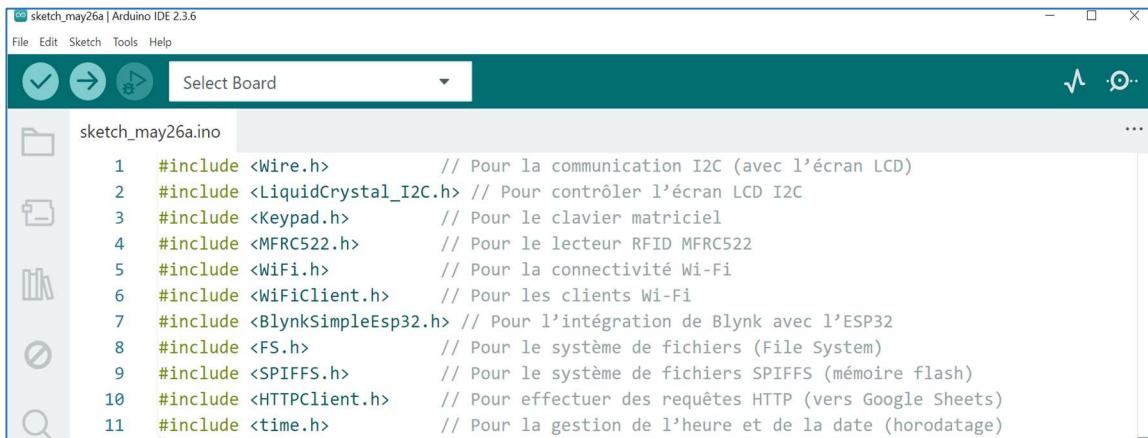
#### 4.1.2. Déroulement du programme

- Déclaration des bibliothèques et des variables
- Exécution de la partie configuration (fonction setup ()) ;
- Le programme boucle sans fin (fonction Loop ()) , exécutant de façon répétée de code

### 4.1.3. Partie déclaration

Inclusions des librairies nécessaires :

Une bibliothèque est un élément essentiel. Elle permet d'exploiter facilement différents composants électroniques ou de gérer la communication entre ces composants et l'ESP32. Il s'agit de dossiers contenant des fonctions prédéfinies qui simplifient la programmation et facilitent l'utilisation des modules matériels.



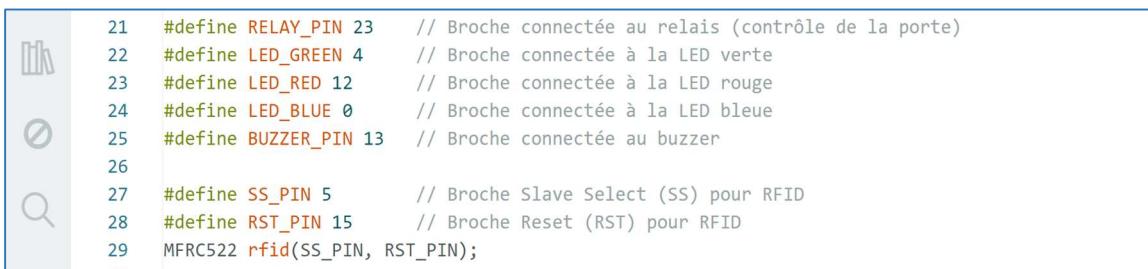
```

sketch_may26a.ino
1 #include <Wire.h>           // Pour la communication I2C (avec l'écran LCD)
2 #include <LiquidCrystal_I2C.h> // Pour contrôler l'écran LCD I2C
3 #include <Keypad.h>          // Pour le clavier matriciel
4 #include <MFRC522.h>         // Pour le lecteur RFID MFRC522
5 #include <WiFi.h>            // Pour la connectivité Wi-Fi
6 #include <WiFiClient.h>       // Pour les clients Wi-Fi
7 #include <BlynkSimpleEsp32.h> // Pour l'intégration de Blynk avec l'ESP32
8 #include <FS.h>              // Pour le système de fichiers (File System)
9 #include <SPIFFS.h>           // Pour le système de fichiers SPIFFS (mémoire flash)
10 #include <HTTPClient.h>        // Pour effectuer des requêtes HTTP (vers Google Sheets)
11 #include <time.h>             // Pour la gestion de l'heure et de la date (horodatage)

```

Figure 41 : Programmation d'ESP32 déclaration des librairies.

Définitions des broches pour les composants physiques :



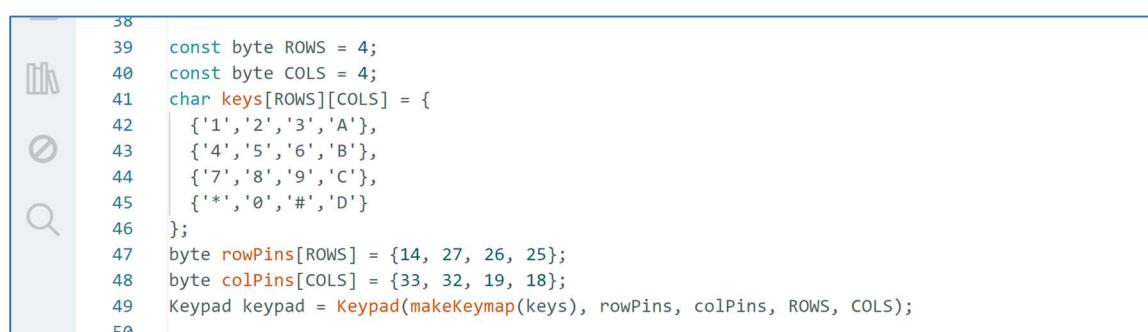
```

21 #define RELAY_PIN 23    // Broche connectée au relais (contrôle de la porte)
22 #define LED_GREEN 4    // Broche connectée à la LED verte
23 #define LED_RED 12     // Broche connectée à la LED rouge
24 #define LED_BLUE 0      // Broche connectée à la LED bleue
25 #define BUZZER_PIN 13    // Broche connectée au buzzer
26
27 #define SS_PIN 5        // Broche Slave Select (SS) pour RFID
28 #define RST_PIN 15       // Broche Reset (RST) pour RFID
29 MFRC522 rfid(SS_PIN, RST_PIN);
30

```

Figure 42 : Programmation d'ESP32 Définitions des broches

Configuration du clavier matriciel :



```

38
39 const byte ROWS = 4;
40 const byte COLS = 4;
41 char keys[ROWS][COLS] = {
42     {'1','2','3','A'},
43     {'4','5','6','B'},
44     {'7','8','9','C'},
45     {'*','0','#','D'}
46 };
47 byte rowPins[ROWS] = {14, 27, 26, 25};
48 byte colPins[COLS] = {33, 32, 19, 18};
49 Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
50

```

Figure 43 : Programmation d'ESP32 initialisations de clavier

Déclaration des variables d'état du système :

```

53  int failedAttempts = 0;
54  bool locked = false;
55  unsigned long lockTime = 0;
56
57  unsigned long ledBuzzerStartTime = 0;
58  unsigned long relayStartTime = 0;
59  bool ledBuzzerOn = false;
60  bool relayOn = false;
61
62  String pendingLogName = "";
63  bool shouldLogAccess = false;
64

```

Figure 44 : Programmation d'ESP32 : déclaration des variables.

Information d'identification pour le mode Point d'Accès Wi-Fi (AP) et pour la connexion Wi-Fi à un réseau existant :

```

54
55  const char* ssid_ap = "ESP32-LOCK";
56  const char* password_ap = "12345678";
57  int previousClientCount = 0;
58
59  const char* ssid = "SABIR";
60  const char* password = "12345678";
61
62  unsigned long lastWifiConnectionAttempt = 0;
63  const unsigned long wifiReconnectInterval = 10000;
64  bool wifiConnected = false;
65

```

Figure 45 : Programmation d'ESP32 Information d'identification

Prototypes des fonctions utilisées dans le programme (déclarations avant leur définition) :

```

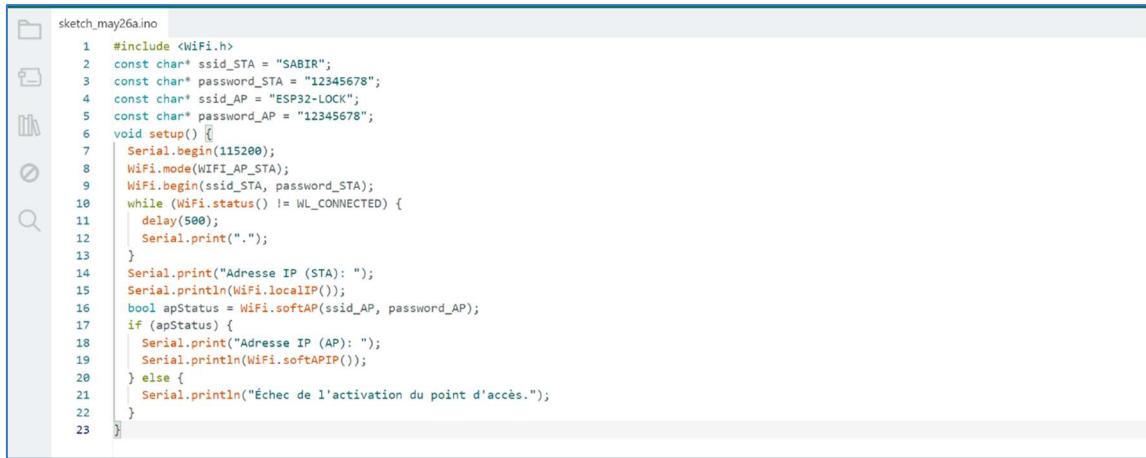
75
76  void openDoor(String name);
77  void checkCode(String code);
78  void checkKeypad();
79  void checkRFID();
80  void lockSystem();
81  void unlockSystem();
82  void handleRelayTiming();
83  void resetDisplay();
84  void centerPrint(String text, int line);
85  void manageWifiConnection();
86  void logAccess(String name);
87  void storeOfflineAccess(String name);
88  void sendStoredAccesses();
89  String getFormattedTime();
90

```

Figure 46 : Programmation d'ESP32 fonctions utilisées

#### 4.1.4. Activation de wifi

Cette section met en évidence la manière dont l'ESP32 peut fonctionner avec une flexibilité remarquable, incarnant simultanément deux rôles vitaux : en tant que point d'accès (AP) qui crée son propre réseau pour fournir un canal de communication direct avec d'autres appareils, et en tant que station (STA) qui connecte le système à un réseau Internet existant ou à un réseau local. Cette double capacité est fondamentale pour la réalisation d'un contrôle à distance efficace, d'une surveillance continue du système, et assure également le flux de données via Internet.



```

1 #include <WiFi.h>
2 const char* ssid_STA = "SABIR";
3 const char* password_STA = "12345678";
4 const char* ssid_AP = "ESP32-LOCK";
5 const char* password_AP = "12345678";
6 void setup() {
7     Serial.begin(115200);
8     WiFi.mode(WIFI_AP_STA);
9     WiFi.begin(ssid_STA, password_STA);
10    while (WiFi.status() != WL_CONNECTED) {
11        delay(500);
12        Serial.print(".");
13    }
14    Serial.print("Adresse IP (STA): ");
15    Serial.println(WiFi.localIP());
16    bool apStatus = WiFi.softAP(ssid_AP, password_AP);
17    if (apStatus) {
18        Serial.print("Adresse IP (AP): ");
19        Serial.println(WiFi.softAPIP());
20    } else {
21        Serial.println("Échec de l'activation du point d'accès.");
22    }
23 }

```

Figure 47 : Programmation d'ESP32 Activation de WIFI

#### 4.1.5. Activation de Blynk avec esp32

La plateforme Blynk est une interface essentielle pour notre système de contrôle d'accès intelligent. Elle permet une surveillance et un contrôle à distance via une application smartphone, connectée à l'ESP32 via Wi-Fi.

Étapes de Connexion à Blynk :

Le dispositif se connecte à Blynk en utilisant un identifiant unique ("Auth Token") et une fois une connexion Wi-Fi stable établie.

Étape de Création de Projet Blynk sur la Console Web :

Pour connecter notre appareil ESP32 à la plateforme, il est d'abord nécessaire de créer un modèle (Template) pour l'appareil sur la Blynk Console. Cette opération constitue la première étape cruciale pour définir le comportement et les interactions de notre dispositif intelligent.

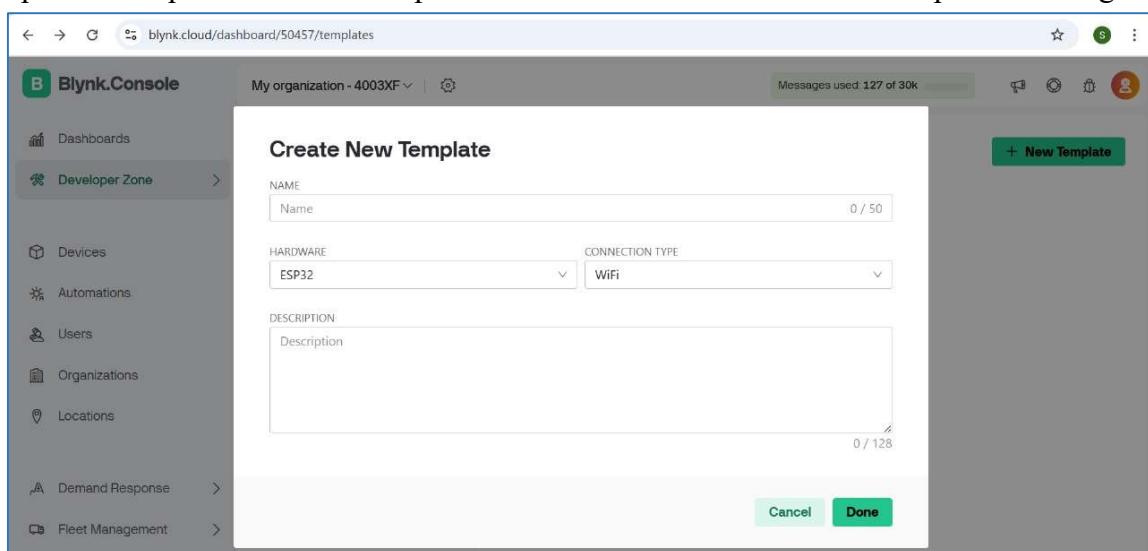


Figure 48 : Écran de création d'un nouveau modèle sur la Blynk Console

Cette image illustre le point de départ de la configuration de notre projet sur Blynk. Nous définissons ici les paramètres fondamentaux du modèle, tels que le nom du projet (e.g. "ON IN

OFF"), le matériel utilisé, spécifiquement l'ESP32, et le type de connexion, qui est le Wi-Fi pour notre dispositif. Ces informations sont cruciales car elles dictent comment l'ESP32 communiquera avec la plateforme Blynk.

Une fois le modèle de base créé, l'étape suivante consiste à configurer les flux de données (Datastreams). Ces flux agissent comme des canaux de communication entre l'ESP32 et la Blynk Console, permettant le transfert bidirectionnel d'informations (par exemple, l'état d'un capteur ou une commande vers un actionneur).

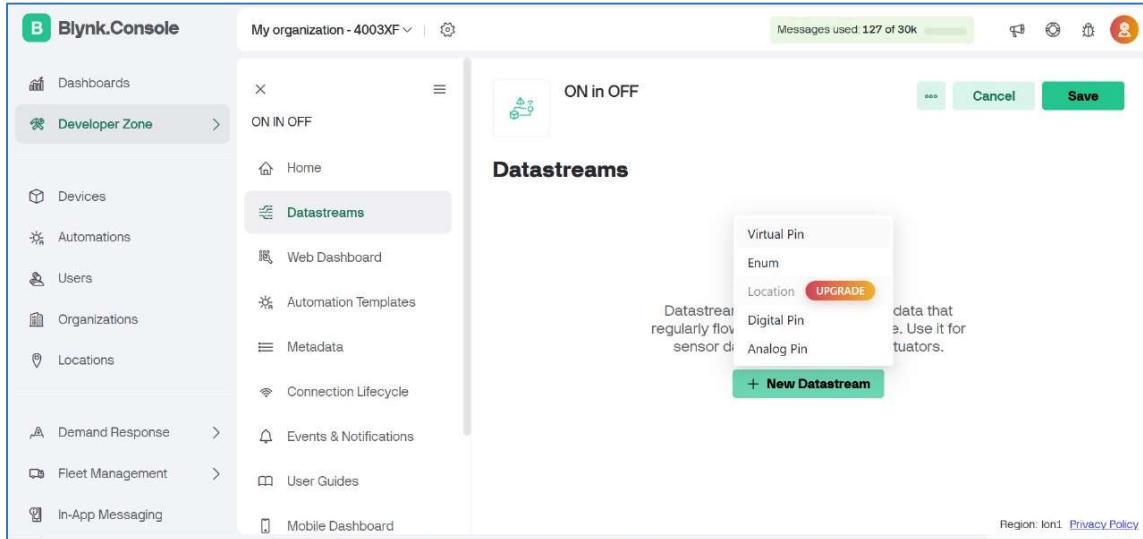


Figure 49 : Page de configuration des Datastreams dans la Blynk Console

Cette figure montre l'interface où nous pouvons ajouter de nouveaux Datastreams. Nous sélectionnons le type de Datastream le plus approprié pour notre application. Dans notre cas, nous privilégions le "Virtual Pin" (broche virtuelle) pour sa flexibilité, permettant d'associer des fonctions logiques spécifiques dans le code de l'ESP32 à des éléments visuels ou des contrôles dans l'application Blynk.

La définition d'un Datastream de type "Virtual Pin" nécessite des précisions sur son comportement et le type de données qu'il manipulera.

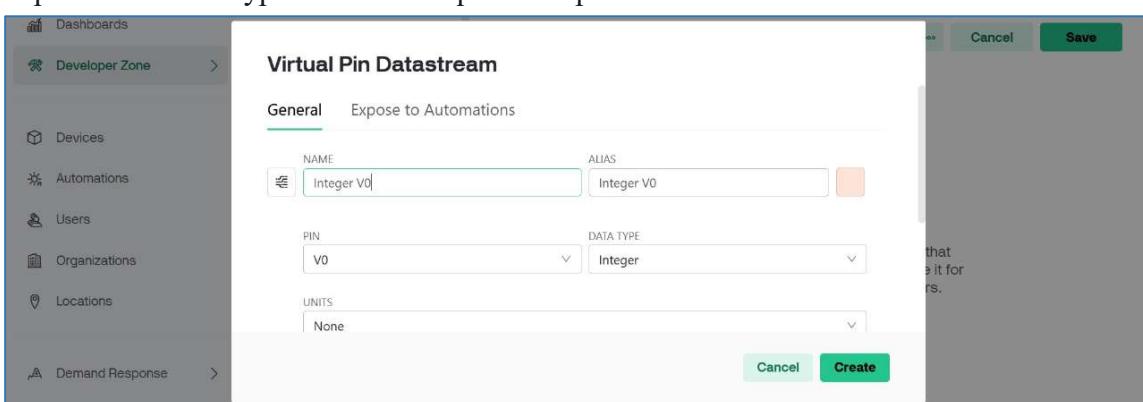


Figure 50 : Fenêtre de configuration d'un Virtual Pin Datastream.

Cette image détaille la configuration spécifique d'un "Virtual Pin Datastream". Nous attribuons un nom clair au Datastream (par exemple, "Integer V0" comme suggéré), définissons le PIN virtuel (ici, V0), et choisissons le type de données ("Integer" pour des valeurs entières comme 0 ou 1, représentant "off" ou "on"). Cette configuration assure que les données échangées entre l'ESP32 et Blynk sont interprétées correctement, permettant un contrôle précis du système.

Avant la connexion: L'application mobile Blynk indique visuellement que le dispositif est hors ligne. Vous pouvez voir le nom du projet, mais une icône spécifique signale l'absence de connexion active avec l'ESP32.

Après la connexion: Une fois la connexion Wi-Fi et Blynk établie, le dispositif est en ligne et prêt à interagir avec l'application.

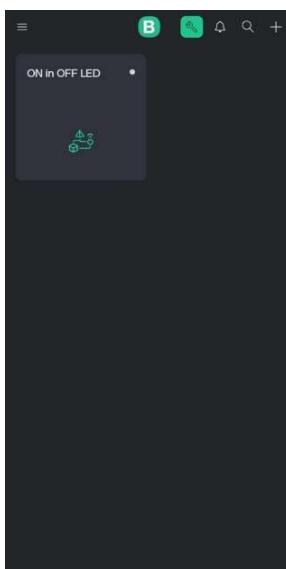


Figure 51 : L'interface Blynk avant la connexion de l'ESP32, montrant l'état hors ligne

Interaction via l'Application Blynk :

Blynk utilise des "broches virtuelles" pour permettre la communication bidirectionnelle entre l'application et l'ESP32, ce qui permet de contrôler les fonctions du système et de visualiser leur état.

Contrôle du verrou (état "Ouvert"): Initialement, après que l'ESP32 se soit connecté à Blynk, le bouton de contrôle du verrou apparaît en rouge avec le texte "off". Dans notre système, cela signifie que le verrou est ouvert.

Contrôle du verrou (état "Fermé"): Lorsque l'utilisateur touche le bouton dans l'application Blynk (passant de "off" à "on"), l'ESP32 reçoit l'ordre et le verrou se ferme. Visuellement, le bouton change de couleur pour devenir vert avec le texte "on".

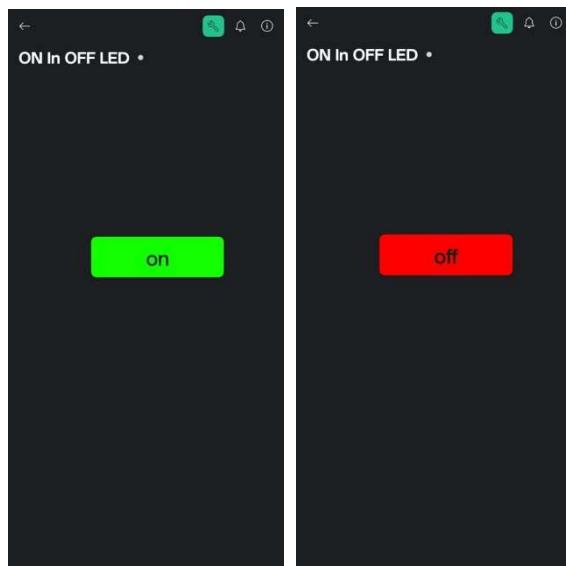


Figure 52 : L'état "on et off" du bouton dans Blynk

#### 4.1.6. Intégration de Google Sheets

Pour enregistrer les informations d'accès comme le nom de l'utilisateur et la date de manière simple et organisée, nous avons connecté l'ESP32 à Google Sheets. Cette méthode nous permet de stocker chaque tentative d'accès dans un tableau en ligne, ce qui facilite le suivi et la consultation des données à tout moment.

Voici les étapes que nous avons suivies pour faire ce lien :

##### 4.1.6.1. Création d'une feuille Google Sheets

Nous avons d'abord créé une nouvelle feuille sur Google Sheets, avec deux colonnes Nom : pour écrire le nom de l'utilisateur.

Time : pour enregistrer la date et l'heure.

	time
1	name
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	

Figure 53 : Création d'une feuille Google Sheets

#### 4.1.6.2. Écriture du Script Google Apps

Nous sommes allés dans : Extensions > Apps Script

Puis nous avons supprimé le code par défaut et ajouté le code suivant :



```

1  function doPost(e) {
2    var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
3
4    var data = JSON.parse(e.postData.contents);
5
6    var name = data.name;
7    var time = data.time;
8
9    sheet.appendRow([name, time]);
10
11   return ContentService.createTextOutput("Success");
12 }

```

Figure 54 : code Google Apps Script

Ce script permet de recevoir les données envoyées par l'ESP32 et de les ajouter dans la feuille.

#### 4.1.6.3. Publication du script en tant que Web App

Après avoir écrit le script, nous l'avons publié comme suit :

Cliquer sur Deploy > Manage deployments > New deployment.

Choisir Web App.

Exécuter en tant que : Moi-même

Qui a accès : Tout le monde

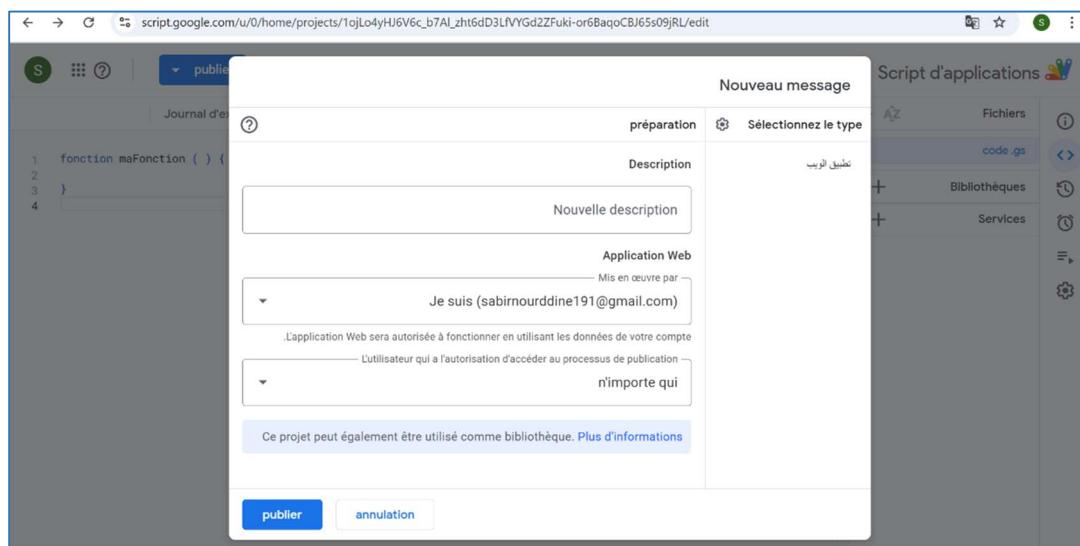


Figure 55 : Publication du script

Une fois publié, nous avons obtenu un lien (URL) que nous avons utilisé dans le code de l'ESP32.

## 4.2. Réalisation physique

### 4.2.1. Montage global

Montage global de système :

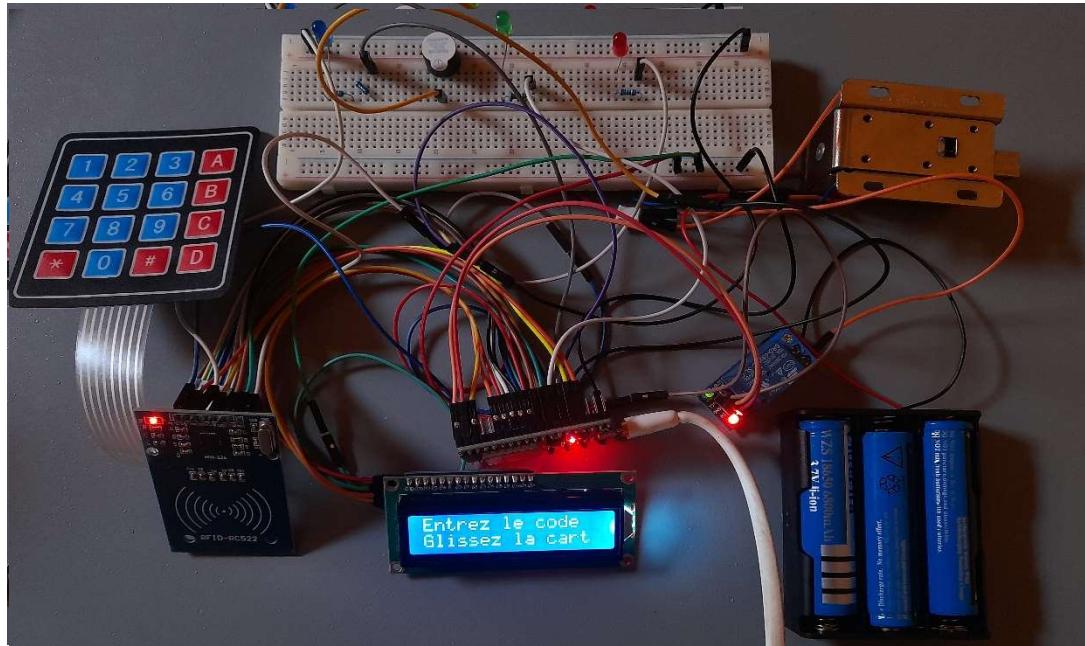


Figure 56 : montage global

### 4.2.2. Résultats pratiques

Accès par badge RFID :

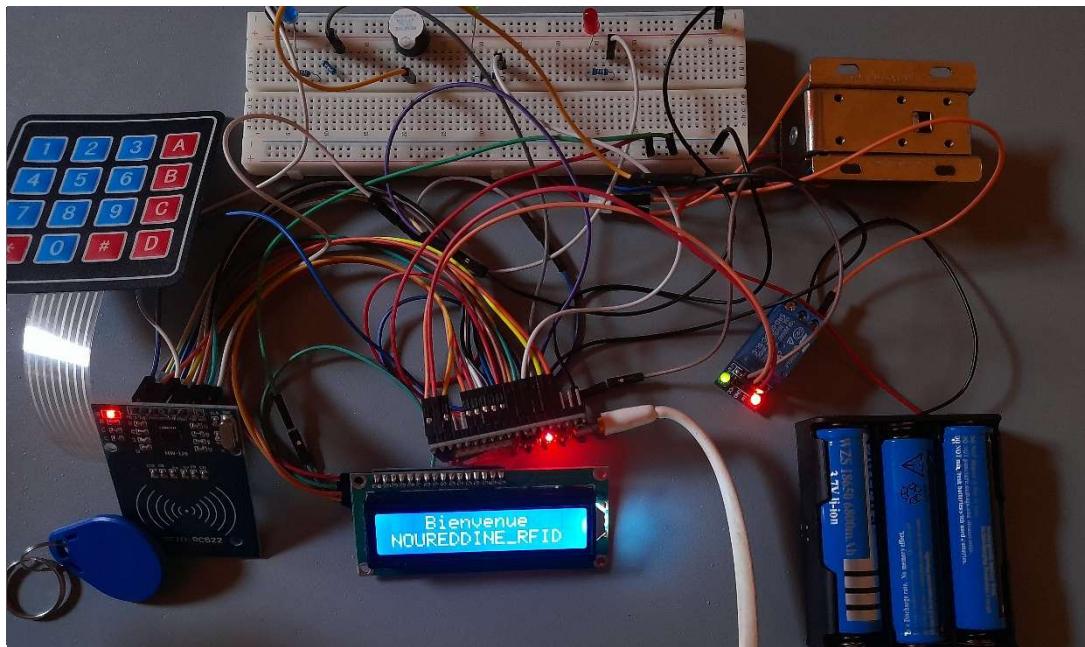


Figure 57 : Accès par badge RFID

Accès par code PIN :

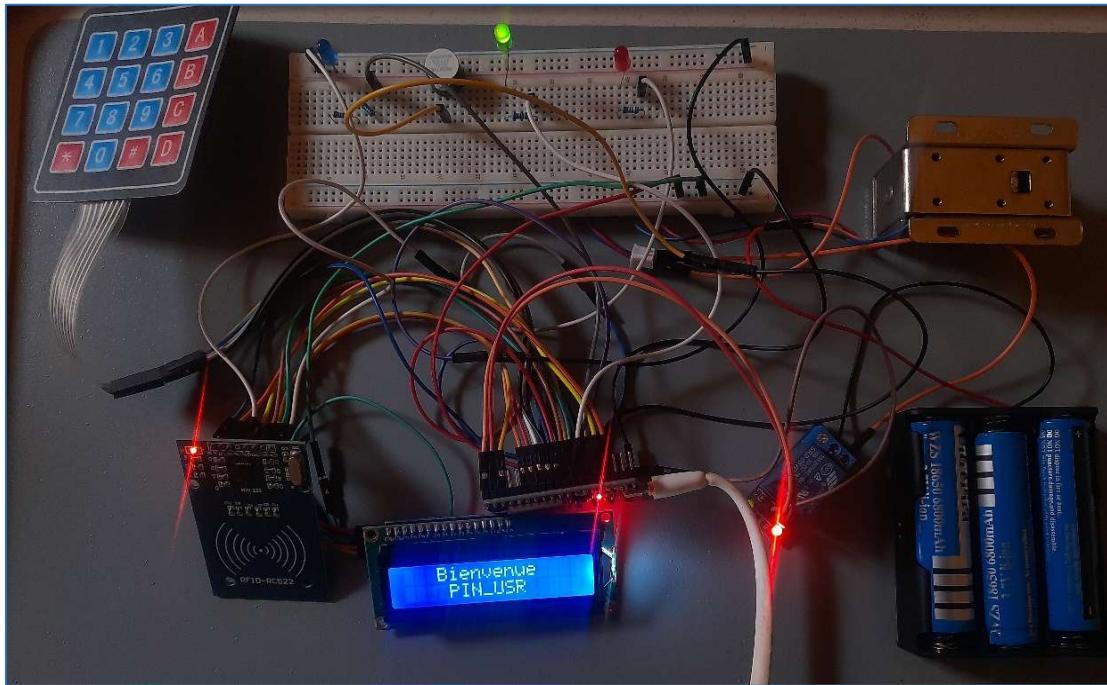


Figure 58 : Accès par code PIN

Accès par point d'accès Wi-Fi :

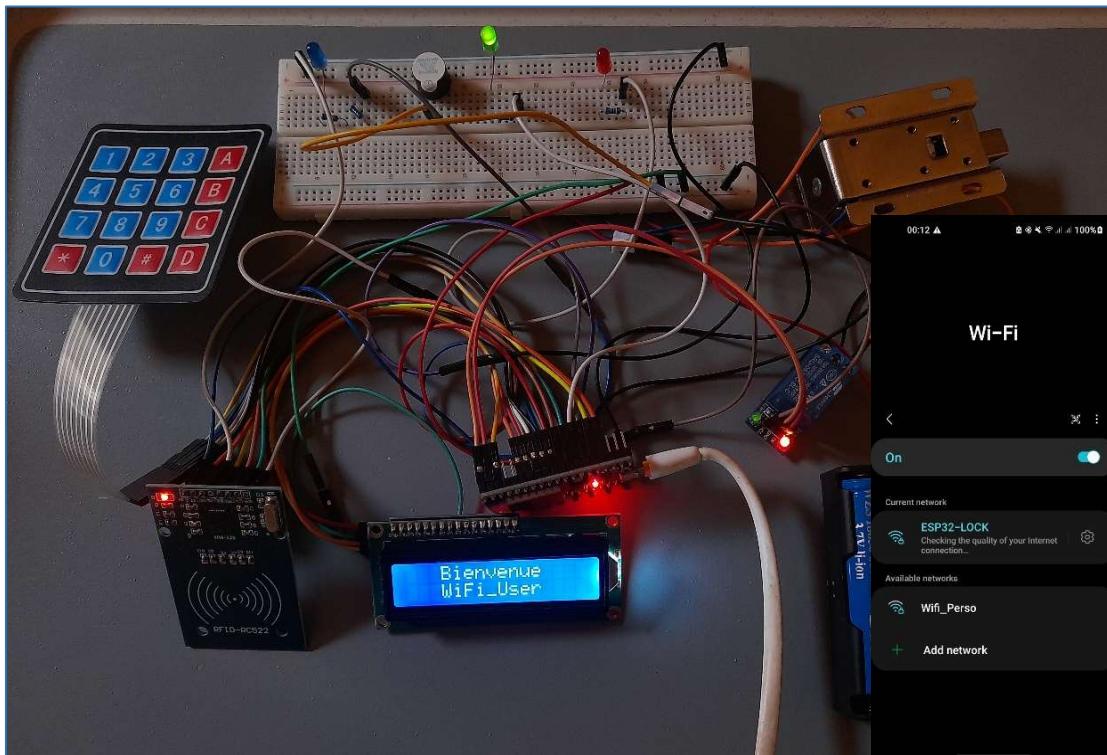


Figure 59 : Accès par point d'accès Wi-Fi

Accès via application Blynk :

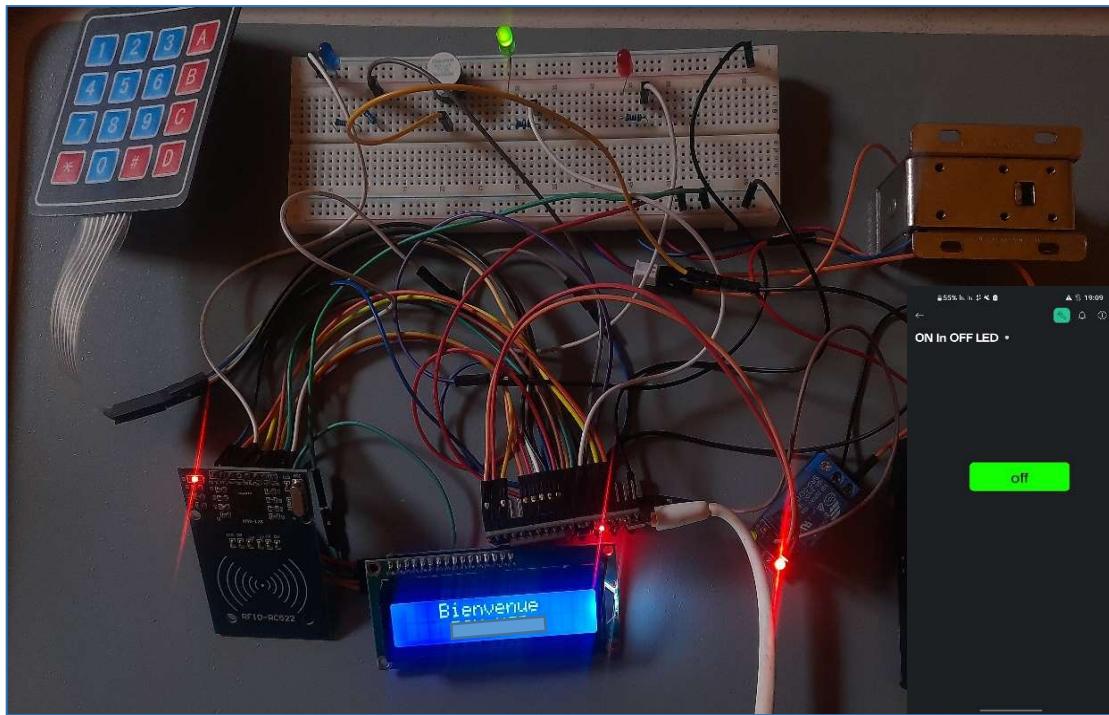


Figure 60 : Accès via application Blynk

Résultats dans Google Sheets :

nom	time
1 name	time
2 NOUREDDINE_RFI	2025-05-23 19:13:57
3 Sabir	2025-05-23 19:14:07
4 Hind	2025-05-23 19:14:15
5 Blynk_User	2025-05-23 19:14:27
6 Omaima	2025-05-23 19:14:38
7 WiFi_User	2025-05-23 19:14:47
8 Omaima	2025-05-23 19:14:57
9 Hind	2025-05-23 19:15:08
10 Mohamad	2025-05-23 19:15:14
11 NOUREDDINE_RFI	2025-05-23 19:15:21
12 Mohamad	2025-05-23 19:15:27
13 Mohamad	2025-05-23 19:15:34
14 Blynk_User	2025-05-23 19:15:42
15 Omaima	2025-05-23 19:16:04
16	
17	
18	

Figure 61 : Résultats dans Google Sheets

#### 4.2.3. Présentation finale du système

Vue frontale du prototype :

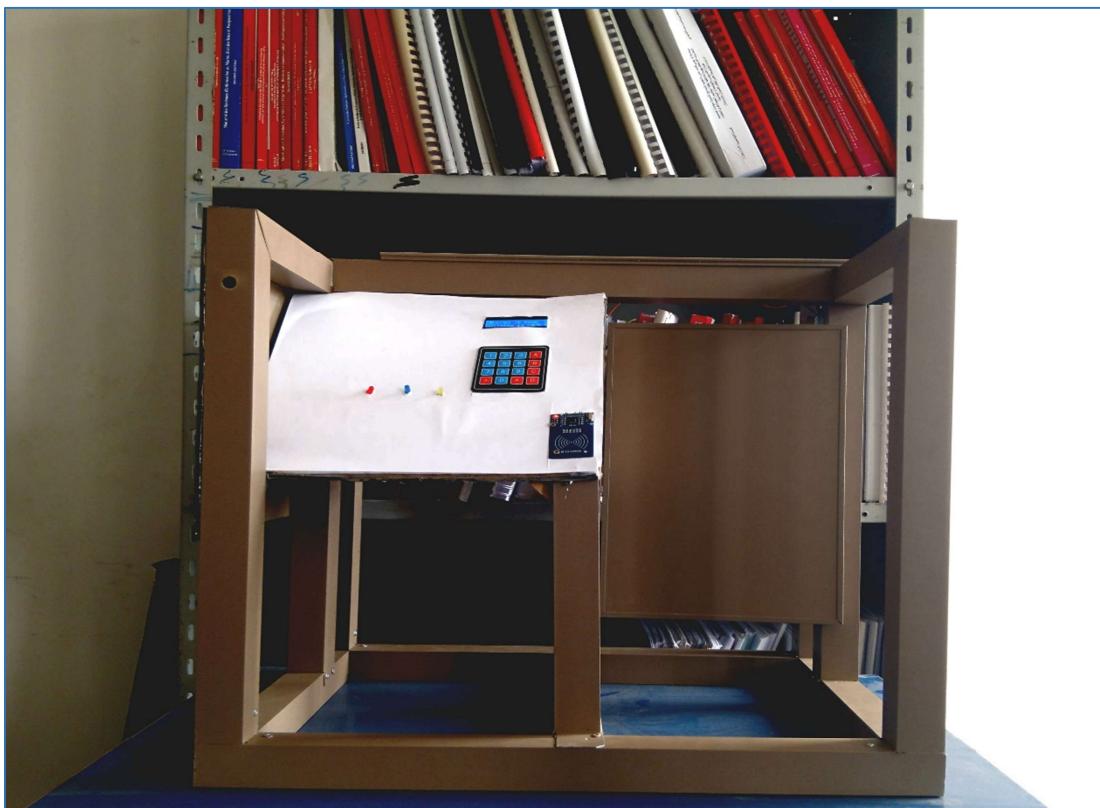


Figure 62 : Vue frontale du prototype

Vue arrière du prototype :

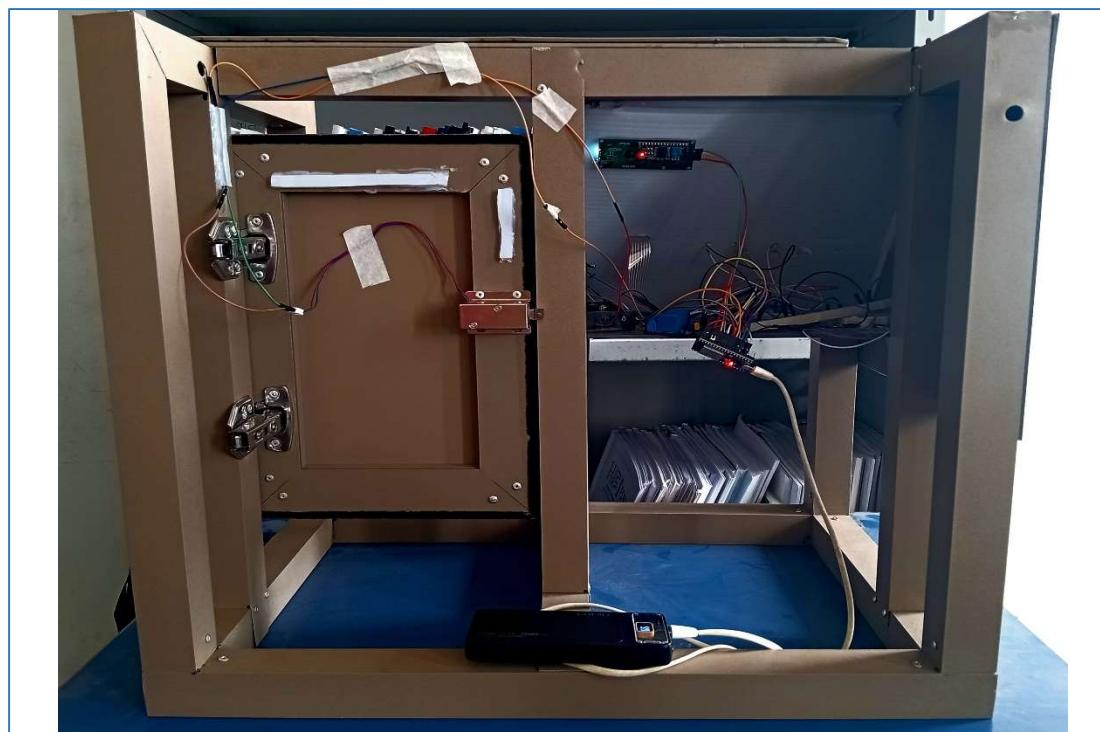


Figure 63 : Vue arrière du prototype

## 5. Conclusion

En conclusion de ce chapitre, nous avons détaillé l'intégralité du processus de conception et de réalisation de notre serrure électronique intelligente. Nous avons d'abord passé en revue les choix architecturaux et les spécifications techniques qui ont guidé le développement du système, justifiant la sélection des composants matériels clés tels que le microcontrôleur ESP32, le lecteur RFID et le servomoteur. S'en est suivi une présentation de l'architecture logicielle, où nous avons détaillé l'intégration des différentes bibliothèques et la logique de fonctionnement de chaque module (gestion des accès par code PIN et RFID, contrôle du relais et communication avec la plateforme Blynk).

La phase de réalisation a permis de transformer les concepts théoriques en un prototype fonctionnel, confirmant la justesse de l'assemblage des composants et de l'application du code. Bien que cette étape ait rencontré certains défis techniques inhérents au développement des systèmes embarqués, les solutions adoptées ont abouti à un système opérationnel, conforme aux exigences fonctionnelles initialement définies.

## Conclusion Générale

## Conclusion Générale

Ce projet est conception d'un système de verrouillage électronique intelligent qui combine de nombreuses méthodes d'authentification modernes telles qu'un code PIN, une technologie et un contrôle RFID et via l'application de téléphone mobile via Wi-Fi .

Pendant le travail, nous avons exploré différentes techniques de verrouillage électronique et choisi les composants des matériaux corrects, puis simuler les fils physiques et le système ordinaire à travers l'environnement Arduino IDE. Le système final répond aux exigences de sécurité, à la simplicité d'utilisation, à la communication et au stockage des informations du système; Stockage de toutes les opérations avec la période pendant laquelle chaque opération a eu lieu

Ce projet nous a également permis d'unifier nos compétences en électronique, sur la programmation disponible et l'intégration des applications à l'avenir. Perspectives d'amélioration du projet; Créez notre propre application et stimulez Bluetooth dans le système de verrouillage entre l'application et l'ESP32 afin de faciliter le travail du système et peut être utilisé par tout le monde

## Annexe

## Annexe 1

### Programme de la cart esp32

```

#define BLYNK_TEMPLATE_ID "TMPL5TRtHte50"
#define BLYNK_TEMPLATE_NAME "ON in OFF LED"
#define BLYNK_AUTH_TOKEN "KuMkEkWSU01tg7IX1eo1KNh7cs4yfFBL"

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <SPI.h>
#include <MFRC522.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <HTTPClient.h>
#include <time.h>
#include <FS.h>
#include <SPIFFS.h>

#define RELAY_PIN 23
#define LED_GREEN 4
#define LED_RED 12
#define LED_BLUE 0
#define BUZZER_PIN 13
#define SS_PIN 5
#define RST_PIN 15

// Google Sheets script URL
const char* googleScriptUrl =
"https://script.google.com/macros/s/AKfycbw2ut88qPf8a_DFc0IMsDr-4912hkJdeC0aRQKB60gKxS6MfdWt25fjTKMyvV6fJWKj/exec";

MFRC522 rfid(SS_PIN, RST_PIN);

String rfids[] = {"F32B729A", "53CFF5DA", "12121212", "34343434"};
String rfidUsers[] = {"NOUREDDINE_RFID", "HIND_RFID", "MOHAMMED_RFID",
"OMAIMA_RFID"};

String pinCodes[] = {"1111", "5555", "8888", "0000", "2222"};
String pinUsers[] = {"Sabir", "Mohamad", "Hind", "Omaima", "PIN_USR"};

LiquidCrystal_I2C lcd(0x27, 16, 2);

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'}
};

byte rowPins[ROWS] = {14, 27, 26, 25};
byte colPins[COLS] = {33, 32, 19, 18};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

String enteredCode = "";

int failedAttempts = 0;
bool locked = false;
unsigned long lockTime = 0;

unsigned long ledBuzzerStartTime = 0;
unsigned long relayStartTime = 0;
bool ledBuzzerOn = false;
bool relayOn = false;

String pendingLogName = "";
bool shouldLogAccess = false;

const char* ssid_ap = "ESP32-LOCK";
const char* password_ap = "12345678";
int previousClientCount = 0;

const char* ssid = "SABIR";
const char* password = "12345678";

unsigned long lastWifiConnectionAttempt = 0;
const unsigned long wifiReconnectInterval = 10000;
bool wifiConnected = false;

void openDoor(String name);
void checkCode(String code);
void checkKeypad();
void checkRFID();
void lockSystem();
void unlockSystem();
void handleRelayTiming();
void resetDisplay();
void centerPrint(String text, int line);
void manageWiFiConnection();
void logAccess(String name);
void storeOfflineAccess(String name);
void sendStoredAccesses();
String getFormattedTime();

```

```
BLYNK_WRITE(V0) {
    if (locked) return;
    int pinValue = param.asInt();
    if (pinValue == 1) {
        openDoor("Blynk_User");
    }
}

void setup() {
    Serial.begin(115200);
    if(!SPIFFS.begin(true)){
        Serial.println("An Error has occurred while mounting SPIFFS");
    }
    pinMode(RELAY_PIN, OUTPUT);
    pinMode(LED_GREEN, OUTPUT);
    pinMode(LED_RED, OUTPUT);
    pinMode(LED_BLUE, OUTPUT);
    pinMode(BUZZER_PIN, OUTPUT);
    pinMode(0, OUTPUT);
    digitalWrite(RELAY_PIN, HIGH);
    digitalWrite(LED_GREEN, LOW);
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_BLUE, LOW);
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(0, LOW);
    lcd.init();
    lcd.backlight();
    resetDisplay();
    SPI.begin(16, 2, 17, 5);
    rfid.PCD_Init();
    configTime(0, 0, "pool.ntp.org", "time.nist.gov");
    WiFi.mode(WIFI_AP_STA);
    WiFi.softAP(ssid_ap, password_ap);
    Serial.print("AP IP address: ");
    Serial.println(WiFi.softAPIP());
    WiFi.begin(ssid, password);
    lastWifiConnectionAttempt = millis();
}

void loop() {
    if (locked) {
        if (millis() - lockTime >= 30000) {
            unlockSystem();
        } else {
            manageWiFiConnection();
            if (wifiConnected) {
                Blynk.run();
            }
        }
    }
}
```

```

        return;
    }
}
checkKeypad();
checkRFID();
int currentClientCount = WiFi.softAPgetStationNum();
if (currentClientCount > previousClientCount) {
    Serial.println("WiFi client connected - open door ");
    openDoor("WiFi_User");
}
previousClientCount = currentClientCount;
handleRelayTiming();
manageWiFiConnection();
if (wifiConnected) {
    Blynk.run();
}
}

void manageWiFiConnection() {
if (WiFi.status() != WL_CONNECTED) {
    wifiConnected = false;
    if (millis() - lastWifiConnectionAttempt >= wifiReconnectInterval) {
        Serial.println("Attempting to reconnect WiFi...");
        WiFi.disconnect();
        WiFi.begin(ssid, password);
        lastWifiConnectionAttempt = millis();
    }
} else {
    if (!wifiConnected) {
        wifiConnected = true;
        Serial.println("WiFi connected");
        Blynk.config(BLYNK_AUTH_TOKEN);
        Blynk.connect();
        sendStoredAccesses();
    }
}
}

void centerPrint(String text, int line) {
lcd.setCursor(0, line);
lcd.print("          ");
lcd.setCursor((16 - text.length()) / 2, line);
lcd.print(text);
}

void resetDisplay() {
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Entrez le code");
}

```

```

lcd.setCursor(0, 1);
lcd.print("Glissez la cart");
}

void checkKeypad() {
    if (locked) return;
    char key = keypad.getKey();
    if (!key) return;
    if (key >= '0' && key <= '9') {
        if (enteredCode.length() < 4) {
            enteredCode += key;
            lcd.setCursor(0, 1);
            lcd.print(enteredCode + "    ");
        }
    } else if (key == '#') {
        if (enteredCode.length() == 4) checkCode(enteredCode);
        enteredCode = "";
    } else if (key == '*') {
        if (enteredCode.length() > 0) {
            enteredCode.remove(enteredCode.length() - 1);
            lcd.setCursor(0, 1);
            lcd.print(enteredCode + "    ");
        }
    }
}
}

void checkRFID() {
    if (locked) return;
    if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) return;
    String uid = "";
    for (byte i = 0; i < rfid.uid.size; i++) {
        char buffer[3];
        sprintf(buffer, "%02X", rfid.uid.uidByte[i]);
        uid += buffer;
    }
    Serial.print("RFID UID: ");
    Serial.println(uid);
    for (int i = 0; i < sizeof(rfids) / sizeof(rfids[0]); i++) {
        if (uid == rfids[i]) {
            openDoor(rfidUsers[i]);
            rfid.PICC_HaltA();
            rfid.PCD_StopCrypto1();
            return;
        }
    }
    lcd.clear();
    centerPrint("RFID inconnu", 0);
    digitalWrite(LED_RED, HIGH);
    digitalWrite(BUZZER_PIN, HIGH);
}

```

```

delay(1000);
digitalWrite(LED_RED, LOW);
digitalWrite(BUZZER_PIN, LOW);
resetDisplay();
failedAttempts++;
if (failedAttempts >= 5) {
    lockSystem();
}
rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
}

void checkCode(String code) {
    for (int i = 0; i < sizeof(pinCodes)/sizeof(pinCodes[0]); i++) {
        if (code == pinCodes[i]) {
            openDoor(pinUsers[i]);
            failedAttempts = 0;
            return;
        }
    }
    failedAttempts++;
    digitalWrite(LED_RED, HIGH);
    digitalWrite(BUZZER_PIN, HIGH);
    lcd.clear();
    centerPrint("Code", 0);
    centerPrint("incorrect", 1);
    delay(1000);
    digitalWrite(LED_RED, LOW);
    digitalWrite(BUZZER_PIN, LOW);
    resetDisplay();
    if (failedAttempts >= 5) {
        lockSystem();
    }
}

void openDoor(String name) {
    lcd.clear();
    centerPrint("Bienvenue", 0);
    centerPrint(name, 1);
    digitalWrite(LED_GREEN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH);
    ledBuzzerStartTime = millis();
    ledBuzzerOn = true;
    pendingLogName = name;
    shouldLogAccess = true;
    Blynk.virtualWrite(V0, 1);
}

void lockSystem() {

```

```

lcd.clear();
centerPrint("Système bloqué", 0);
centerPrint("30 secondes", 1);
digitalWrite(LED_BLUE, HIGH);
digitalWrite(0, HIGH);
lockTime = millis();
locked = true;
failedAttempts = 0;
if (wifiConnected) {
    Blynk.virtualWrite(V0, 0);
}
}

void unlockSystem() {
    locked = false;
    digitalWrite(LED_BLUE, LOW);
    digitalWrite(0, LOW);
    resetDisplay();
}

void handleRelayTiming() {
    if (ledBuzzerOn && millis() - ledBuzzerStartTime >= 1000) {
        digitalWrite(LED_GREEN, LOW);
        digitalWrite(BUZZER_PIN, LOW);
        ledBuzzerOn = false;
        digitalWrite(RELAY_PIN, LOW);
        relayStartTime = millis();
        relayOn = true;
        if (shouldLogAccess) {
            if (wifiConnected) {
                logAccess(pendingLogName);
            } else {
                storeOfflineAccess(pendingLogName);
            }
            shouldLogAccess = false;
        }
    }
    if (relayOn && millis() - relayStartTime >= 3000) {
        digitalWrite(RELAY_PIN, HIGH);
        relayOn = false;
        resetDisplay();
        if (wifiConnected) {
            Blynk.virtualWrite(V0, 0);
        }
    }
}

void logAccess(String name) {
    String timeStr = getFormattedTime();

```

```

if (timeStr == "") timeStr = "No Time";
String jsonData = "{\"name\":\"" + name + "\", \"time\":\"" + timeStr +
"}";
HTTPClient http;
http.begin(googleScriptUrl);
http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(jsonData);
if (httpResponseCode <= 0) {
    Serial.print("Error sending to Google Sheets: ");
    Serial.println(httpResponseCode);
    storeOfflineAccess(name);
}
http.end();
}

void storeOfflineAccess(String name) {
File file = SPIFFS.open("/access_log.txt", FILE_APPEND);
if (file) {
    file.println(name + "," + String(millis()));
    file.close();
}
}

void sendStoredAccesses() {
if (!SPIFFS.exists("/access_log.txt")) return;
File file = SPIFFS.open("/access_log.txt", FILE_READ);
while (file.available()) {
    String line = file.readStringUntil('\n');
    int commaIndex = line.indexOf(',');
    if (commaIndex == -1) continue;
    String name = line.substring(0, commaIndex);
    String timeStr = "Offline-" + line.substring(commaIndex + 1);
    String jsonData = "{\"name\":\"" + name + "\", \"time\":\"" + timeStr +
"}";
    HTTPClient http;
    http.begin(googleScriptUrl);
    http.addHeader("Content-Type", "application/json");
    if (http.POST(jsonData) > 0) {
        delay(100);
    } else {
        break;
    }
    http.end();
}
file.close();
SPIFFS.remove("/access_log.txt");
}

String getFormattedTime() {

```

```

struct tm timeinfo;
if (!getLocalTime(&timeinfo)) {
    return "";
}
char timeString[20];
strftime(timeString, sizeof(timeString), "%Y-%m-%d %H:%M:%S", &timeinfo);
return String(timeString);
}

```

## Annexe 2

### Liste des composants et matérielle nécessaire

Type	Référence	Quantité
Esp32	Wroom 32	1
RFID	RC522	1
Clavier matriciel	4x4	1
LCD I2C	16x2	1
Serrure électromagnétique 12v	12v	1
Relais	Module 5v	1
Buzzer		1
LED		3
Résistance	200 $\Omega$	3
Cable USB	Type AB	1
Plaque d'essai	830 points	1
Batterie	5v	1
Batterie	12v	1

## Références

## Références

- [1] [https://en-m.wikipedia.org.translate.goog/wiki/Electronic\\_lock?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=fr&\\_x\\_tr\\_hl=fr&\\_x\\_tr\\_pto=rq#:~:text=An%20electronic%20lock%20\(or%20electric,mounted%20directly%20to%20the%20lock.](https://en-m.wikipedia.org.translate.goog/wiki/Electronic_lock?_x_tr_sl=en&_x_tr_tl=fr&_x_tr_hl=fr&_x_tr_pto=rq#:~:text=An%20electronic%20lock%20(or%20electric,mounted%20directly%20to%20the%20lock.)
- [2] [En ligne]. Available: <https://www.verisure.fr/comment-fonctionne-une-serrure-connectee#:~:text=Une%20serrure%20connect%C3%A9e%20fonctionne%20gr%C3%A2ce,mobile%20ou%20un%20dispositif%20d%C3%A9di%C3%A9A9>.
- [3] [En ligne]. Available: <https://www.lesbonsartisans.fr/comprendre-le-fonctionnement-des-serrures-electriques/>.
- [4] M. Langheinrich, «A survey of rfid privacy approaches. personal and ubiquitous computing,13,» 2009.
- [5] B. JAMALI, «The evolution of RFID,» Adelaide, australia, 2025.
- [6] «proeedings of the world congreses on engineering and computer science,» san francisco USA, 2009 .
- [7] «RFID: Technology and applicationa sridhar,» bombay.
- [8] [En ligne]. Available: <https://www.lesbonsartisans.fr/la-technologie-nfc-revolution-dans-la-securisation-des-acces#:~:text=La%20technologie%20NFC%20%3A%20d%C3%A9finition%20et,typiquement%20inf%C3%A9rieure%20%C3%A0%2010%20cm.>
- [9] [En ligne]. Available: <https://www.lesbonsartisans.fr/la-technologie-nfc-revolution-dans-la-securisation-des-acces#:~:text=Son%20fonctionnement%20s'appuie%20sur,de%20mani%C3%A8re%20%C3%A0%20curis%C3%A9e%20et%20rapide.>
- [10] [En ligne]. Available: <https://www.intel.fr/content/www/fr/fr/products/docs/wireless/how-does-bluetooth-work.html>.
- [11] «a,» [En ligne]. Available: <https://www.intel.fr/content/www/fr/fr/products/docs/wireless/how-does-bluetooth-work.html>.
- [12] B. KHOUKHA, «Système d'ouverture de porte par empreinte digitale,» 2022.
- [13] [En ligne]. Available: [https://www.verisure.fr/guide-securite/systeme-d-alarme/composants-alarme/detecteur-de-presence/fonctionnement-capteur-infrarouge?utm\\_source=chatgpt.com](https://www.verisure.fr/guide-securite/systeme-d-alarme/composants-alarme/detecteur-de-presence/fonctionnement-capteur-infrarouge?utm_source=chatgpt.com).

- [14] [En ligne]. Available: [https://www.verisure.fr/guide-securite/systeme-d-alarme/composants-alarme/detecteur-de-presence/fonctionnement-capteur-infrarouge?utm\\_source=chatgpt.com](https://www.verisure.fr/guide-securite/systeme-d-alarme/composants-alarme/detecteur-de-presence/fonctionnement-capteur-infrarouge?utm_source=chatgpt.com).
- [15] «serrurier savoyard,» [En ligne]. Available: <https://serrurier-savoyard.fr/serrure-biometrique/>. [Accès le 13 3 2025].
- [16] e. microplanet. [En ligne]. Available: <https://www.micro-planet.ma/produit/esp-32s-dual-core-2-4ghz-wifi-bluetooth/>.
- [17] [En ligne]. Available: <https://www.sac-marquage.com/utilisation-module-rfid-rc522-pour-arduino/>.
- [18] [En ligne]. Available: <https://learn.circuit.rocks/rfid-interfacing-mfrc-522-reader-writer-module>.
- [19] [En ligne]. Available: [https://riflbiometrics.com/tout-ce-que-vous-devez-savoir-sur-le-badge-rfid/#Decouvrez\\_le\\_MIFARE\\_DESFIRE\\_EV2\\_la\\_solution\\_parfaite\\_pour\\_proteger\\_vos\\_locaux\\_et\\_vos\\_donnees](https://riflbiometrics.com/tout-ce-que-vous-devez-savoir-sur-le-badge-rfid/#Decouvrez_le_MIFARE_DESFIRE_EV2_la_solution_parfaite_pour_proteger_vos_locaux_et_vos_donnees).
- [20] R. fonctionnnement. [En ligne]. Available: [https://riflbiometrics.com/tout-ce-que-vous-devez-savoir-sur-le-badge-rfid/#Decouvrez\\_le\\_MIFARE\\_DESFIRE\\_EV2\\_la\\_solution\\_parfaite\\_pour\\_proteger\\_vos\\_locaux\\_et\\_vos\\_donnees](https://riflbiometrics.com/tout-ce-que-vous-devez-savoir-sur-le-badge-rfid/#Decouvrez_le_MIFARE_DESFIRE_EV2_la_solution_parfaite_pour_proteger_vos_locaux_et_vos_donnees).
- [21] [En ligne]. Available: <https://www.robotique.tech/tutoriel/utilisation-du-clavier-matriciel-4x4-avec-arduino/#:~:text=Un%20clavier%20matriciel,et%20les%20colonnes..>
- [22] [En ligne]. Available: <https://www.robotique.tech/tutoriel/utilisation-du-clavier-matriciel-4x4-avec-arduino/#:~:text=Un%20afficheur%20I2C,de%20contr%C3%B4le%20industriels>.
- [23] [En ligne]. Available: <https://www.robotique.tech/tutoriel/utilisation-du-clavier-matriciel-4x4-avec-arduino/#:~:text=Pour%20utiliser%20un,configuration%20et%20d%E2%80%99utilisation>.
- [24] LED. [En ligne]. Available: <https://leclairage.fr/led/>.
- [25] buzzer. [En ligne]. Available: <https://www.alarmemaison.com/content/84-buzzer-piezoelectrique-fonctionnement-avantages-et-applications-dans-les-systemes-d-alarme>.
- [26] c. d. buzzer. [En ligne]. Available: <https://www.moussasoft.com/produit/buzzer-actif-5v/>.

- [27] s. electromagnétique. [En ligne]. Available: <https://edukeytech.com/produit/serrure-electromagnetique-12v/>.
- [28] relais. [En ligne]. Available: <https://www.micro-planet.ma/produit/module-1-relais-5v/>.
- [29] «Baia-Rasso bir«Conception d'un système de gestion d'une cité intelligente» Mémoire De master, universite de BADJI MOKHTAR ANNAB».
- [30] [En ligne]. Available: <https://roboman.in/wp-content/uploads/2022/09/1602-lcd-with-i2c-interface-india-800x800-1.jpg>.