
TABLE DES MATIÈRES

Introduction générale	1
1 Contexte général du projet	3
Introduction	4
1.1 Cadre général du sujet	4
1.2 Présentation du sujet	4
1.3 Étude de l'existant	4
1.3.1 Analyse de l'existant	4
1.3.2 Critique de l'existant	5
1.3.3 Solution proposée	5
1.4 analyse du cahier des charges	5
1.4.1 Pourquoi les serrures électroniques?	6
1.5 Spécifications des besoins	7
1.5.1 les besoins logiciels	7
1.5.2 les besoins materiels :	8
1.5.3 les besoins materiels :	9
Conclusion	12
2 conception du cahier de charge	13
Introduction	14
2.1 Architecture de la solution	14

2.2	Conception détaillée :	15
2.2.1	Diagramme fonctionnel	15
2.2.2	Diagramme de cas d'utilisation	16
2.2.3	Diagramme de séquence :	17
2.2.4	Diagramme de définition de blocs :	18
2.2.5	Diagramme d'états de transitions :	19
	Conclusion	20
3	simulation du projet	21
	Introduction	22
3.1	simulation	22
3.1.1	pourquoi utiliser proteus?	22
3.1.2	Simulation des composants électroniques	23
3.1.3	Simulation des composants logiciels	23
3.1.4	Simulation des scénarios d'utilisation	24
3.2	Limitations de la simulation	26
3.3	Perspectives d'amélioration de la simulation	26
	Conclusion	27
4	Réalisation du projet	28
	Introduction	29
4.1	Environnement de travail :	29
4.1.1	Environnement matériel	29
4.2	montage physique	30
4.3	installation du logiciel :	31
4.4	Programmation :	32
4.5	test de systme :	33
4.6	Intégration et validation :	35
4.7	conception d'interface web :	37
	Conclusion	41
	conclusion générale et perspectives	42

TABLE DES FIGURES

1.1	Schemad'un systeme RFID	6
1.2	Raspberry pi imager	7
1.3	python	8
1.4	html css js	8
1.5	Visual Studio Code	8
1.6	Visual Studio Code	9
1.7	Une diode :	9
1.8	Rfid module	10
1.9	Camera raspberry	10
1.10	Servo moteur	11
1.11	Afficheur LCD I2C	11
1.12	fil électrique	11
1.13	résistance	12
2.1	l'architecture du projet	15
2.2	Diagramme fonctionnel	16
2.3	l'architecture du projet	17
2.4	Diagramme de séquence	18
2.5	Diagramme de définition de blocs	19
2.6	Diagramme d'états de transitions	20
3.1	Simulation des composants électroniques	23
3.2	Simulation des composants logiciels	24

3.3	Senario 1 :carte valide	25
3.4	Senario 2 :carte valide	26
4.1	les Caractéristiques de machine	29
4.2	les Caractéristiques Les terminaux mobiles :	30
4.3	montage physique	31
4.4	montage physique	32
4.5	Programmation	33
4.6	envoi du mail	33
4.7	envoi du mail	34
4.8	test du carte valide	34
4.9	test du carte invalide	35
4.10	attente du badge	35
4.11	cas d'autorisation du carte	36
4.12	cas d'echec du carte	36
4.13	interface web d'accès à distance	37
4.14	Firebase	38
4.15	Connexion avec raspberry	38
4.16	Connexion avec raspberry	39
4.17	twilio	40
4.18	Envoi d'un SMS	40
4.19	Envoi d'un mail	40
4.20	verification	40

INTRODUCTION GÉNÉRALE

L'être humain cherche toujours à mettre en place un système de sécurité et de surveillance fiable afin de protéger ses biens immobiliers et les locaux collectifs contre les intrusions et les prévenir contre le vol. Les serrures ont pour but d'assurer cette tache 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 coté composants (microcontrôleurs programmables, carte Raspberry pi).

Il existe une grande variété de serrures adaptées à tous types de portes et portillons, parmi eux la serrure électrique dont elle est fabriquée en acier renforcé et ne peut pas être percée ou coupée. Cette technologie de fabrication ajoute une grande amélioration par rapport aux serrures traditionnelles. Les anciens Égyptiens ont été les premiers à verrouiller leurs portes à l'aide de serrures à clé. Bien que la technologie des serrures ait toujours évolué, ce n'est que récemment qu'elle est devenue assez intelligente pour se débarrasser complètement de ces clés.

Grâce à la technologie de sécurité intelligente de la maison, nous pouvons maintenant profiter d'un confort sans clé avec une meilleure connaissance et un meilleur contrôle des personnes qui entrent dans nos maisons. Les serrures intelligentes offrent aux propriétaires beaucoup plus qu'un simple changement dans leur sécurité. La nature automatisée de ces serrures offre à ses utilisateurs un niveau de commodité et d'efficacité supplémentaire que les serrures traditionnelles avec cylindres ou bâillet n'offrent pas. Plongeons dans les ser-

rures intelligentes. Les serrures intelligentes remplissent cette fonction traditionnelle, mais elles le font d'une manière très différente, et avec un ensemble d'outils très différent. Ces serrures exécutant les mouvements de verrouillage et de déverrouillage une fois qu'elles ont reçu les instructions appropriées d'un dispositif autorisé, qui utilise un protocole sans fil et une clé cryptographique, assurant ainsi que le processus d'autorisation se déroule correctement.

Le logiciel allant avec la serrure connectée doit être utilisé, dans la plupart des cas, avec un réseau Wifi, un Smartphone ou un hub intelligent. L'aspect logiciel d'une serrure intelligente n'est jamais autonome. Il doit constamment travailler avec d'autres fonctions pour s'assurer qu'il fonctionne au mieux de ses capacités. La disponibilité des microcontrôleurs (PIC, ARM, AVR, ...) et les systèmes embarqués à faible coût comme Raspberry PI, FPGA, BeagleBone..... etc., avec leurs propres plateformes Open Source, est une cause importante des incroyables innovations et inventions électroniques qui ont changé nos vies, en fournissant un large espace de développement avec des possibilités d'implémentation très vastes et une diversité exceptionnelle.

Dans ce rapport, on va proposer une architecture d'un système intelligent sous la forme d'une box domotique à faible coût, qui présente comme éléments intelligents : Une carte Raspberry Pi 4, un lecteur RFID, un afficheur LCD, un servo moteur une camera ,et des leds pour la création d'une authentification a deux facteurs de sécurité qui est la présentation de la carte RFID et l'envoie d'un mail contenant des images en temps réelle et avec un script écrit en langage de programmation Python3 .

Ce rapport est composé de trois chapitres, le premier est une présentation d'un état de l'art sur la technologie actuelle sur les serrures intelligentes, et comme solution on propose et on décrit notre système à faible coût dans un deuxième chapitre ainsi que la description des différents composants utilisés dans ce projet. Le troisième chapitre est dédié à la présentation des étapes de simulation et la réalisation pratique du système, les codes sources, les schémas descriptifs, les modes de fonctionnements, ainsi que la mise en place du verrou réaliser.

CHAPITRE 1

CONTEXTE GÉNÉRAL DU PROJET

Introduction	4
1.1 Cadre général du sujet	4
1.2 Présentation du sujet	4
1.3 Étude de l'existant	4
1.3.1 Analyse de l'existant	4
1.3.2 Critique de l'existant	5
1.3.3 Solution proposée	5
1.4 analyse du cahier des charges	5
1.4.1 Pourquoi les serrures électroniques?	6
1.5 Spécifications des besoins	7
1.5.1 les besoins logiciels	7
1.5.2 les besoins materiels :	8
1.5.3 les besoins materiels :	9
Conclusion	12

Introduction

Dans ce premier chapitre, nous allons analyser le cahier des charges, pour déterminer les exigences logicielles et matérielles (hard et soft), puis nous spécifierons les besoins liés à ces exigences.

1.1 Cadre général du sujet

Ce projet vise a valider la matiere demini projet des systemes microprocesseurs à l'institut superieur d'informatique ariana (isi). Ce projet vise également à renforcer les connaissances théoriques et à les rendre plus utiles et plus développées dans un environnement professionnel.

1.2 Présentation du sujet

L'automatisation des portes est de plus en plus courante dans les bâtiments résidentiels et commerciaux pour offrir un accès sécurisé et pratique. L'ajout de la technologie RFID dans les portes peut améliorer la sécurité en permettant un accès facile et contrôlé aux personnes autorisées.

1.3 Étude de l'existant

L'analyse de l'existant ou analyse détaillée vise à étudier le système existant, le diagnostiquer et déterminer des objectifs du nouveau système et d'ébaucher de solution. Pour ce faire, nous devons étudier le système existant lui-même ainsi que l'environnement dans lequel il baigne.

1.3.1 Analyse de l'existant

L'ouverture de portes traditionnelles se fait généralement à l'aide d'une clé physique ou d'un verrouillage manuel. Cependant, cela peut être problématique en cas de perte de clé ou de clé volée. De plus, la sécurité n'est pas garantie car une personne mal intentionnée peut facilement reproduire une clé physique.

1.3.2 Critique de l'existant

Les portes traditionnelles peuvent être facilement forcées ou piratées, ce qui compromet la sécurité des personnes et des biens. De plus, les portes à clé peuvent être difficiles à gérer dans les bâtiments avec un grand nombre d'utilisateurs. La problématique est de concevoir une porte intelligente qui peut offrir un accès contrôlé et sécurisé, tout en étant facile à gérer pour les propriétaires de bâtiments.

1.3.3 Solution proposée

Le développement de solutions de portes intelligentes avec RFID offre de nombreux avantages, tels que la sécurité accrue, la gestion des accès simplifiée et l'amélioration de l'expérience utilisateur. Ce projet vise à concevoir une porte intelligente avec RFID qui peut être utilisée dans différents types de bâtiments.

1.4 analyse du cahier des charges

Pour définir le plus clairement possible ,Une serrure électronique est un dispositif de verrouillage qui fonctionne à l'aide d'un courant électrique. C'est un dispositif parfois autonome avec un ensemble de commande électronique monté directement sur la serrure, et qui peut être connectés à un système de contrôle d'accès. Les serrures électroniques peuvent également être surveillés à distance et contrôlés, à la fois pour verrouiller et déverrouiller. Le présent chapitre sera consacré à la présentation des principales techniques de verrou électronique.

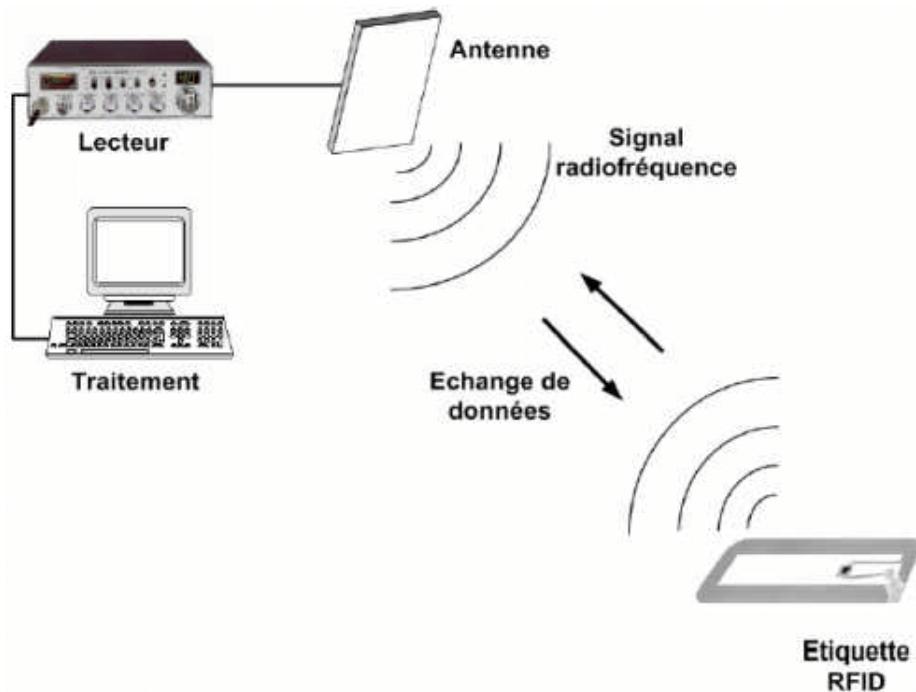


FIGURE 1.1 – Schéma d'un système RFID

1.4.1 Pourquoi les serrures électroniques ?

Le principal intérêt des cylindres électroniques, c'est leur intelligence, qui autorise l'accès Ou le refuse. Dans un organigramme mécanique, une clé peut ouvrir plusieurs portes. Mais le système de contrôle d'accès permet plus de précision et de transparence. Les objectifs de ce projet sont de concevoir, simuler et réaliser une porte intelligente avec RFID qui répond aux exigences suivantes :

- * Permettre l'identification des utilisateurs autorisés à l'aide de la technologie RFID
- * Offrir une sécurité accrue grâce à la technologie RFID et aux mécanismes de verrouillage appropriés
- * Être facile à gérer pour les propriétaires de bâtiments, notamment en permettant la mise à jour facile des accès autorisés
- * Être facile à utiliser pour les utilisateurs, offrant une expérience d'accès pratique et intuitive -Être économiquement viable, en utilisant des composants abordables et en minimisant les coûts de conception et de fabrication.

Ces avantages expliquent l'engouement des propriétaires pour les smart door . Les acteurs présents sur le marché de l'intelligence artificielle l'ont bien compris et proposent des solutions de plus en plus performantes. Ils sont ainsi en mesure de vous

proposer des actions en fonction de vos goûts ou de vos habitudes.

1.5 Spécifications des besoins

Pour assurer un bon fonctionnement et atteindre les performances recherchées dans le cahier des charges, nous avons besoin de certains matériels et certains logiciels décrits dans les lignes qui suivent.

1.5.1 les besoins logiciels

Raspberry pi imager :

Raspberry Pi Imager, destiné à simplifier l'installation des systèmes d'exploitation sur la carte SD du Raspberry Pi . **Raspberry pi imager** : c'est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

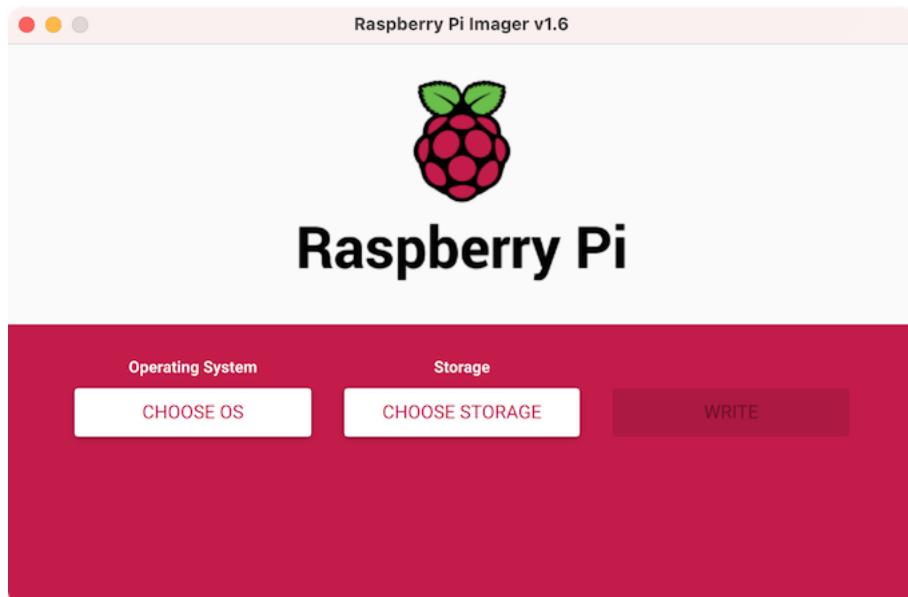


FIGURE 1.2 – Raspberry pi imager

python :

c'est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.



FIGURE 1.3 – python

html css js :

HTML, CSS et JavaScript sont les noms de différents types de code fréquemment rencontrés dans le développement web. L'HTML permet l'affichage des informations de la page sur le navigateur, le CSS met en page son contenu et JavaScript intègre les animations et spécifie les interactions entre les éléments.



FIGURE 1.4 – html css js

Visual Studio Code :

est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.



FIGURE 1.5 – Visual Studio Code

1.5.2 les besoins matériels :

Le Raspberry Pi :

Pi est un micro-ordinateur mono carte à processeur ARM basé sur Linux conçu par un professeur du département informatique de l'Université de Cambridge dans le cadre de la Fondation Raspberry Pi.



FIGURE 1.6 – Visual Studio Code

1.5.3 les besoins matériels :

Une diode :

Une diode : électroluminescente (abrégé en DEL en français, ou LED, de l'anglais : light-emitting diode) est un dispositif opto-électronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.



FIGURE 1.7 – Une diode :

Rfid module :

Le module est un lecteur de la puce RFID basé sur le circuit MFRC522 à faible coût est facile à utiliser. C'est une carte d'interface compatible Arduino et Raspberry Pi, basée sur le circuit MFRC-522 de NXP et est utilisée pour lire et écrire sur des cartes ou badges RFID de type Mifare.



FIGURE 1.8 – Rfid module

Camera raspberry :

Le Raspberry Pi Camera Module v2 est une carte complémentaire conçue sur mesure pour Raspberry Pi, avec un objectif fixe. Il est capable d'images statiques 3280 x 2464 pixels, et prend également en charge les vidéos.



FIGURE 1.9 – Camera raspberry

Servo moteur :

Servo moteur : Un servomoteur (souvent abrégé en « servo », provenant du latin servus qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort

statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure.



FIGURE 1.10 – Servo moteur

Afficheur LCD I2C :

L'écran I2C LCD 1602A permet d'afficher simplement quelques informations au format texte. Cet afficheur est en particulier une interface visuelle entre un système (projet) et l'homme(utilisateur).



FIGURE 1.11 – Afficheur LCD I2C

fil électrique : qui relie les dipôles d'un circuit entre eux.



FIGURE 1.12 – fil électrique

résistance : contrôler l'intensité du courant dans les différents segments d'un circuit.



FIGURE 1.13 – résistance

Conclusion

Dans ce chapitre, nous avons exposé les différentes exigences du cahier des charges, les différentes raisons qui encouragent à opter une porte intelligente aussi que les besoins que ce soit matériels ou logiciels pour la réalisation de ce projet . Dans le prochain chapitre, nous traiterons de manière détaillée la conception du cahier des charges.

CHAPITRE 2

CONCEPTION DU CAHIER DE CHARGE

Introduction	14
2.1 Architecture de la solution	14
2.2 Conception détaillée:	15
2.2.1 Diagramme fonctionnel	15
2.2.2 Diagramme de cas d'utilisation	16
2.2.3 Diagramme de séquence :	17
2.2.4 Diagramme de définition de blocs:	18
2.2.5 Diagramme d'états de transitions:	19
Conclusion	20

Introduction

Dans cette partie de notre travail, nous allons commencer par présenter l'architecture fonctionnelle de notre projet puis nous aborderons la conception en utilisant les diagrammes du langage de modélisation des systèmes. Le contexte du projet nous amène vers les technologies spécifiques à l'ingénierie des systèmes complexes. Ainsi, notre choix est porté sur le SysML un langage de modélisation des systèmes. Il permet avec ses diagrammes, de spécifier, d'analyser et de concevoir un système en toute fluidité.

2.1 Architecture de la solution

Le système que nous allons concevoir dans les lignes suivantes est muni d'un bloc permettant d'actionner un servo moteur , et un bloc pour un circuit complémentaire(les capteurs).

L'architecture d'un projet de smart door avec RFID peut varier en fonction des spécifications et des fonctionnalités souhaitées, mais voici une architecture générale possible :

RFID Reader : Un module RFID qui permet de lire les cartes RFID. Microcontrôleur : Un microcontrôleur comme Arduino ou Raspberry Pi pour contrôler et traiter les données de la RFID Reader. Communication sans fil : Un module sans fil, comme Wi-Fi ou Bluetooth, pour permettre la communication avec d'autres appareils. Alimentation électrique : Une source d'alimentation électrique pour alimenter l'ensemble du système.

Système de déverrouillage de la porte : Un système de déverrouillage de la porte, qui peut être un servomoteur ou un actionneur, pour ouvrir et fermer la porte.

Logiciel : Un programme informatique qui contrôle le système et effectue des tâches telles que la vérification des identifiants RFID et le déverrouillage de la porte.

Base de données : Une base de données qui stocke les informations d'identification des utilisateurs et des autres données pertinentes pour le système.

La figure 2.1 représente l'architecture du notre projet.

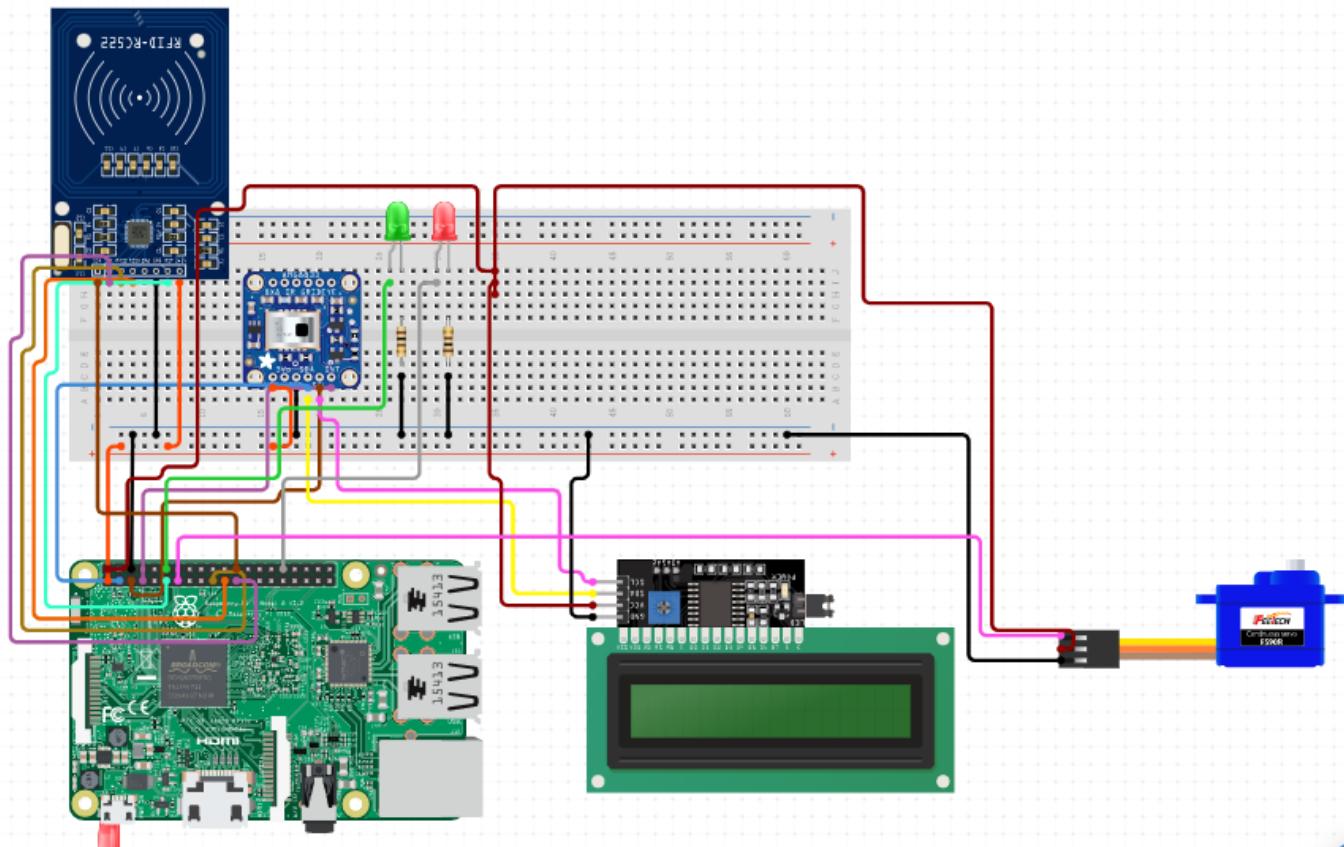


FIGURE 2.1 – l'architecture du projet

2.2 Conception détaillée :

Dans cette partie nous allons détailler la conception du projet en citons des diagrammes

2.2.1 Diagramme fonctionnel

Avec ce diagramme, nous vous expliquons ce que doit faire notre système. Il représente le cahier de charge fonctionnel ou les contraintes de notre système. La carte est chargée d'actionner un servo moteur et capturer des images après la réception des grandeurs physiques par les capteurs. Pour réaliser ce besoin, nous aurons besoin d'un servo moteur, camera, leds, des valeurs fournies par les capteurs et une alimentation énergétique, la figure 2.2 récapitule les exigences de la carte.

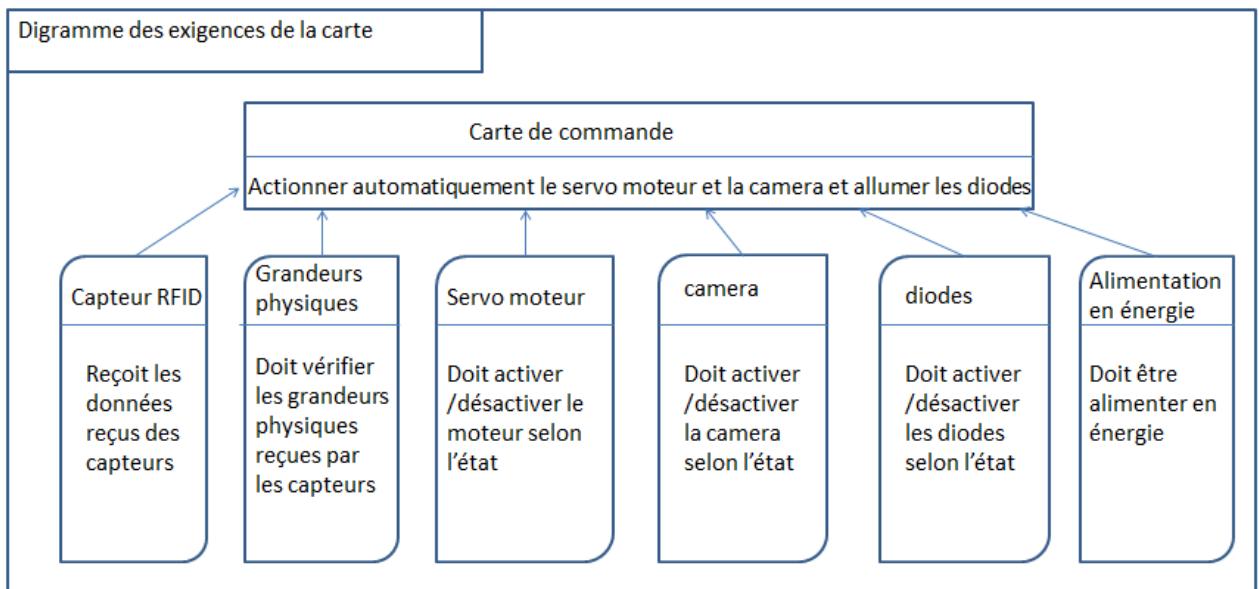


FIGURE 2.2 – Diagramme fonctionnel

2.2.2 Diagramme de cas d'utilisation

Dans ce diagramme, nous vous expliquons les services offerts par la carte aux acteurs. La carte commence par s'alimenter à partir d'une d'énergie puis elle envoie un message au Gateway. Ce message peut être une identification dans le réseau ou une demande des variables au serveur d'application ou un envoi d'acquittement après une opération. Ensuite, elle attend la réponse au message envoyé si celui-ci est une demande de décision d'actionnement. Une fois les décisions reçues, notre solution peut alors soit actionner le servo moteur ou commander d'autres circuits. La figure 2.3 énumère ces actions.

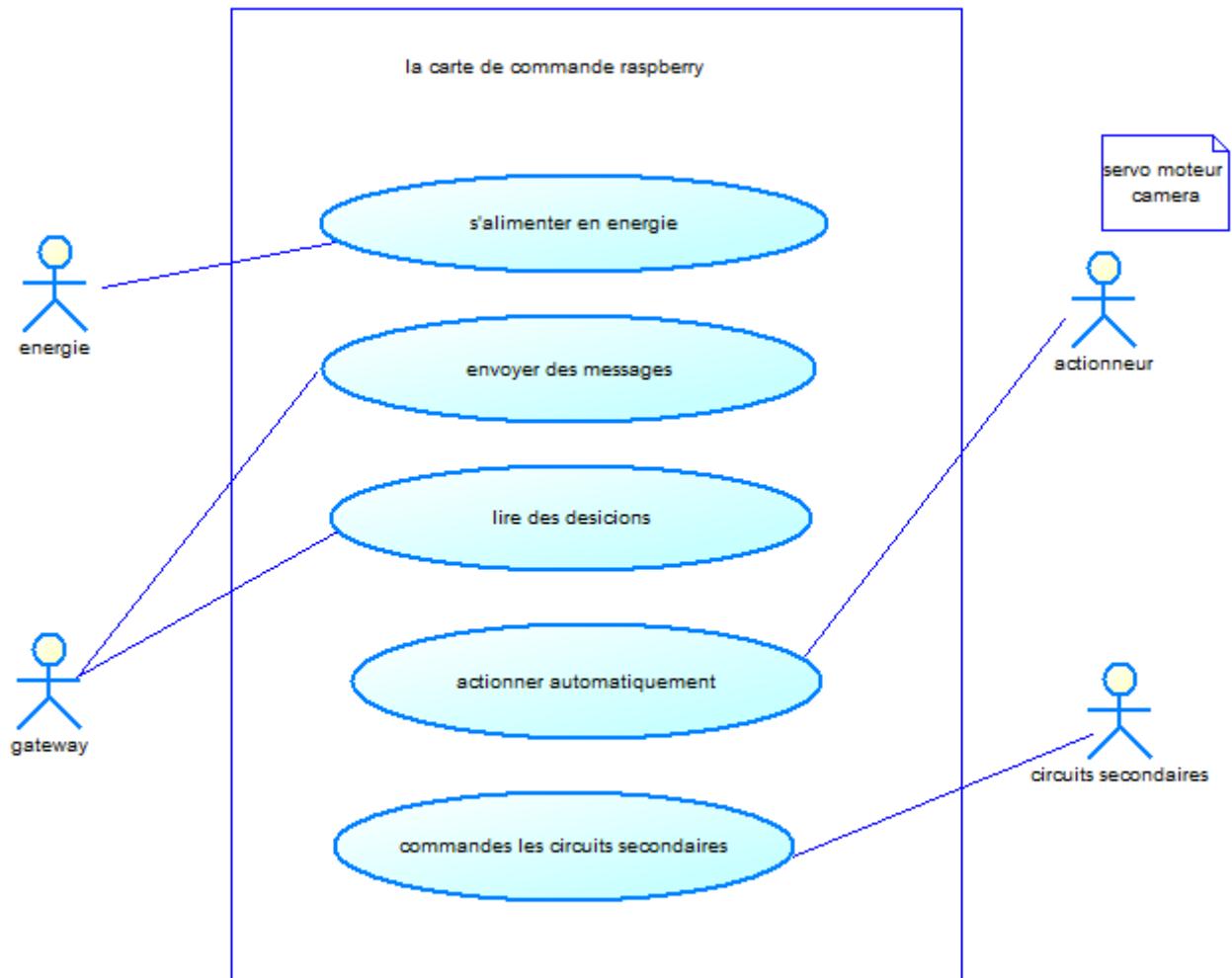


FIGURE 2.3 – l'architecture du projet

2.2.3 Diagramme de séquence :

La figure 2.4 explique l'enchaînement chronologique des différentes interactions entre la carte et les composants environnents. La carte se met sous tension, envoie un message au Gateway et attend la réponse durant une certaine période. Si la réponse est bien reçue, la active ses sorties concernées issues de la réponse.

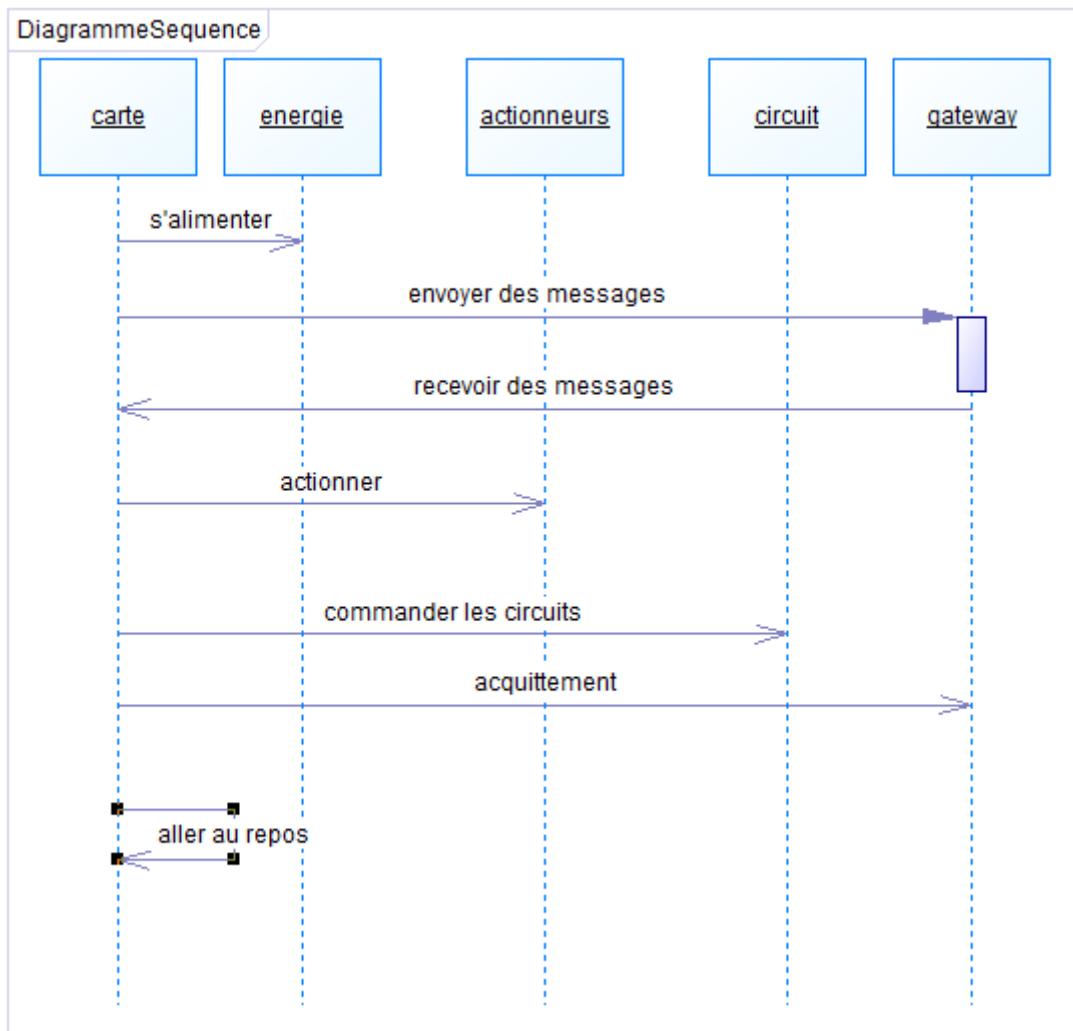


FIGURE 2.4 – Diagramme de séquence

2.2.4 Diagramme de définition de blocs :

Dans ce diagramme, nous avons identifié les différents blocs qui constituent la solution. Il est observé que la carte commence par s'alimenter. Cela nécessite un bloc alimentation. Puis elle doit faire traitement comme affecter le contenu de la réponse au message envoyer aux variables, activer au moins une sortie. Toutes opérations nécessitent un bloc de contrôle et télécommunications. Pour activer la sortie de commande de servo moteur et camera demande un bloc consisté Gateway. un bloc pour la commande des circuits de détection est aussi nécessaire. Enfin, un bloc d'envoie des mails qui récupère des informations . La figure 2.5 illustre la composition de la carte.

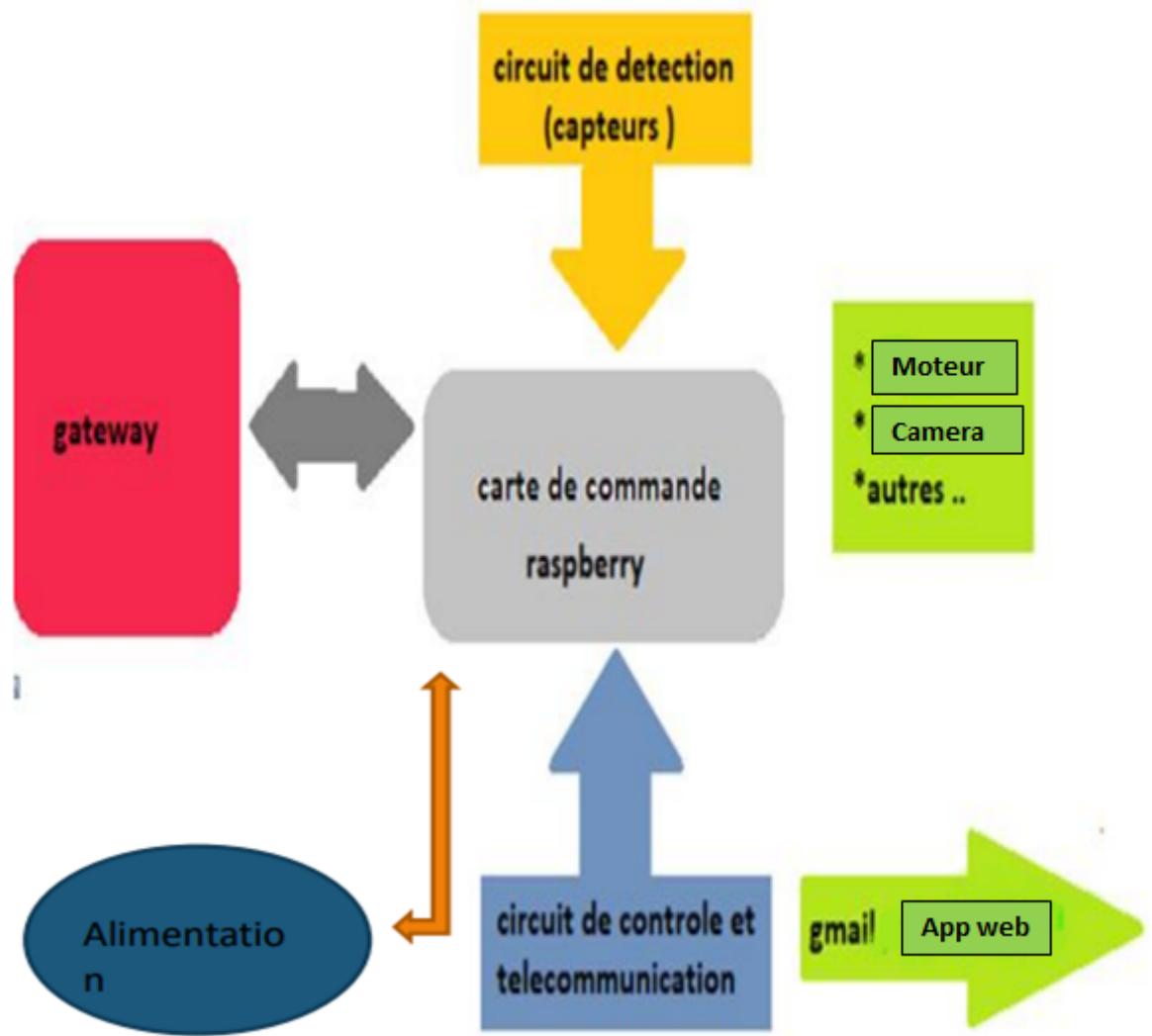


FIGURE 2.5 – Diagramme de définition de blocs

2.2.5 Diagramme d'états de transitions :

Dans cette partie de la conception, nous nous sommes concentrés à élaborer les différents états possibles que peut prendre notre système une fois mis en marche. Au départ, le système est mis au repos donc il reste à l'état, puis elle envoie un message demande des variables au Gateway ainsi elle passe à l'état d'attente de la réponse. Si la réponse n'est pas reçue à temps, elle se met au repos (contrainte de réaction en temps réel). Une fois la bonne réponse est reçue à temps, il se met à l'état d'affectation des décisions aux différentes variables prévues. Quand le temps de cette affectation est écoulé, le système passe à l'état de planification des actions dans lequel toutes les actions comme actionner le système d'alarme, commander le circuit supplémentaire. Enfin, le système repart au repos.

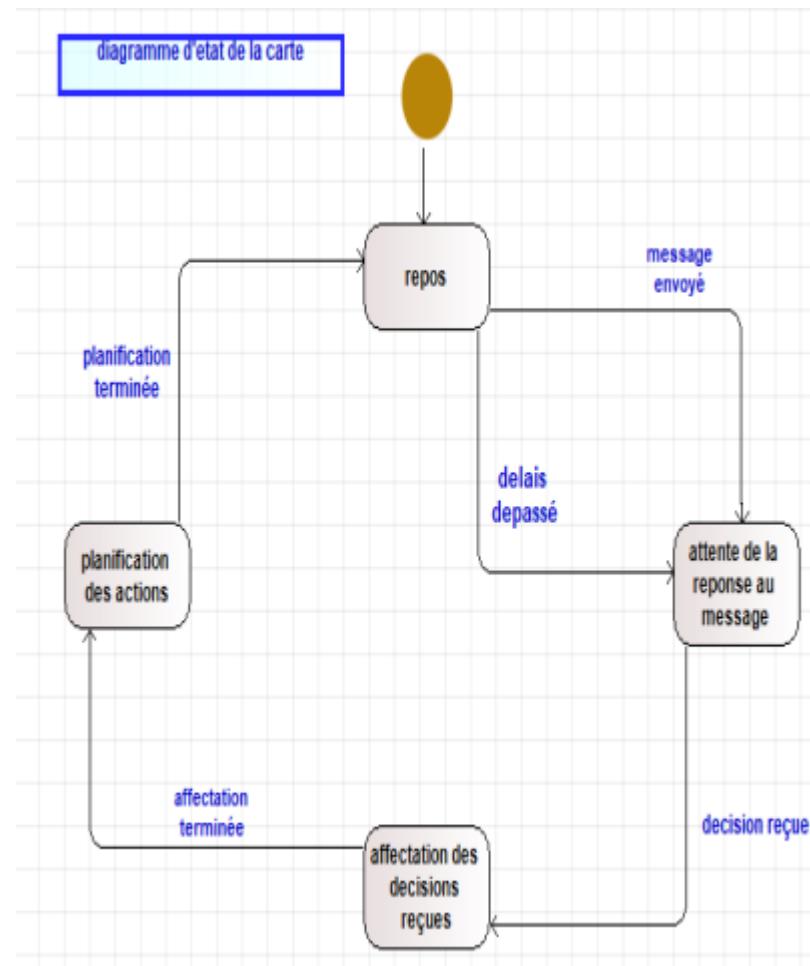


FIGURE 2.6 – Diagramme d’états de transitions

Conclusion

Ce chapitre s'est articulé sur la présentation de l'architecture fonctionnelle de la solution ainsi que la conception détaillée en utilisant les diagrammes du langage de modélisation de système. Le chapitre suivant sera consacré à la réalisation de cette conception

CHAPITRE 3

SIMULATION DU PROJET

Introduction	22
3.1 simulation	22
3.1.1 pourquoi utiliser proteus?	22
3.1.2 Simulation des composants électroniques	23
3.1.3 Simulation des composants logiciels	23
3.1.4 Simulation des scénarios d'utilisation	24
3.2 Limitations de la simulation	26
3.3 Perspectives d'amélioration de la simulation	26
Conclusion	27

Introduction

Dans ce chapitre, nous allons commencer par présenter notre simulation de porte intelligente ,avec un logiciel de simulation.

3.1 simulation

Nous avons utilisé le logiciel de simulation Proteus pour simuler le circuit électronique utilisé pour contrôler la porte intelligente. Nous avons choisi Proteus pour sa facilité d'utilisation et sa capacité à simuler les signaux électriques et électroniques. La figure 2.1 représente l'architecture du notre projet.

3.1.1 pourquoi utiliser proteus ?

Proteus est une plateforme de simulation électronique largement utilisée par les ingénieurs électroniques pour la conception, la simulation et la vérification des circuits électroniques. Nous avons choisi d'utiliser Proteus pour simuler notre projet de porte intelligente avec RFID car cela nous a offert plusieurs avantages.

Tout d'abord, Proteus est une plateforme de simulation très intuitive et facile à utiliser. Il dispose d'une interface utilisateur conviviale qui permet aux utilisateurs de concevoir et de simuler rapidement leurs circuits électroniques sans avoir besoin de connaissances approfondies en programmation ou en conception de circuits.

En outre, Proteus propose une large gamme de composants électroniques simulés, ce qui nous a permis de trouver tous les composants nécessaires pour simuler notre projet de porte intelligente avec RFID. Cela a permis une simulation précise du fonctionnement de notre projet, ce qui a permis de détecter et de corriger les erreurs de conception avant de passer à la phase de prototypage.

Enfin, Proteus est une plateforme de simulation très fiable qui nous a permis d'économiser du temps et de l'argent en évitant les erreurs coûteuses de conception. En simulant notre projet sur Proteus, nous avons pu valider notre conception et garantir son fonctionnement correct avant de passer à la phase de prototypage.

En somme, Proteus a été un choix judicieux pour simuler notre projet de porte intelligente avec RFID en raison de sa facilité d'utilisation, de sa gamme de composants électroniques simulés et de sa fiabilité.

3.1.2 Simulation des composants électroniques

Nous avons inclus les composants suivants dans notre simulation : Raspberry Pi, module RFID, servo-moteur, diodes, résistances et camera. Nous avons interconnecté ces composants selon le schéma électrique de notre circuit. Nous avons vérifié que les signaux étaient correctement transmis entre les différents composants en utilisant le simulateur de signal de Proteus.

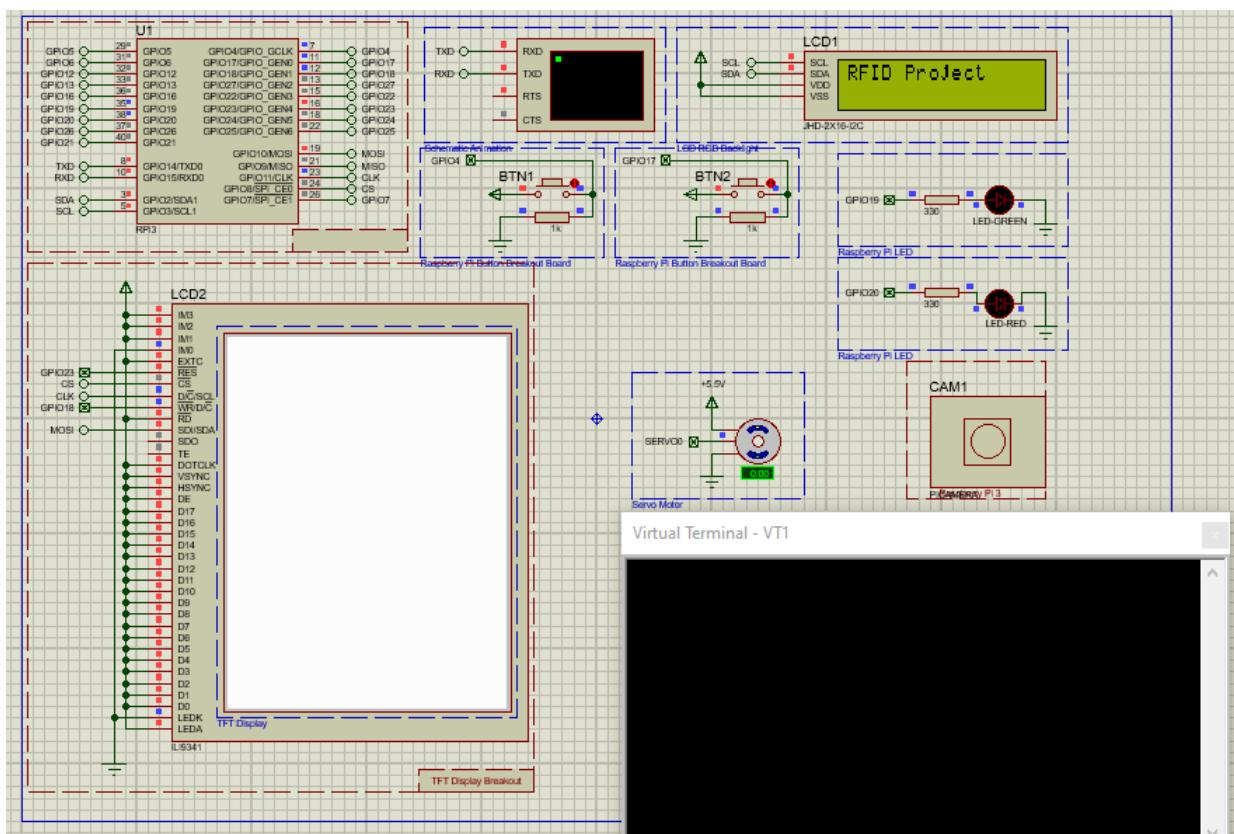


FIGURE 3.1 – Simulation des composants électroniques

3.1.3 Simulation des composants logiciels

Nous avons simulé le code source utilisé pour contrôler les différents composants de la porte intelligente en utilisant le simulateur de code source de Proteus. Nous avons simulé les différentes étapes impliquées dans la reconnaissance de la carte RFID, l'ouverture de la porte, la capture d'images de la caméra et la commande du servo-moteur. Nous avons

vérifié que le code était fonctionnel et qu'il était capable de répondre aux différentes situations.

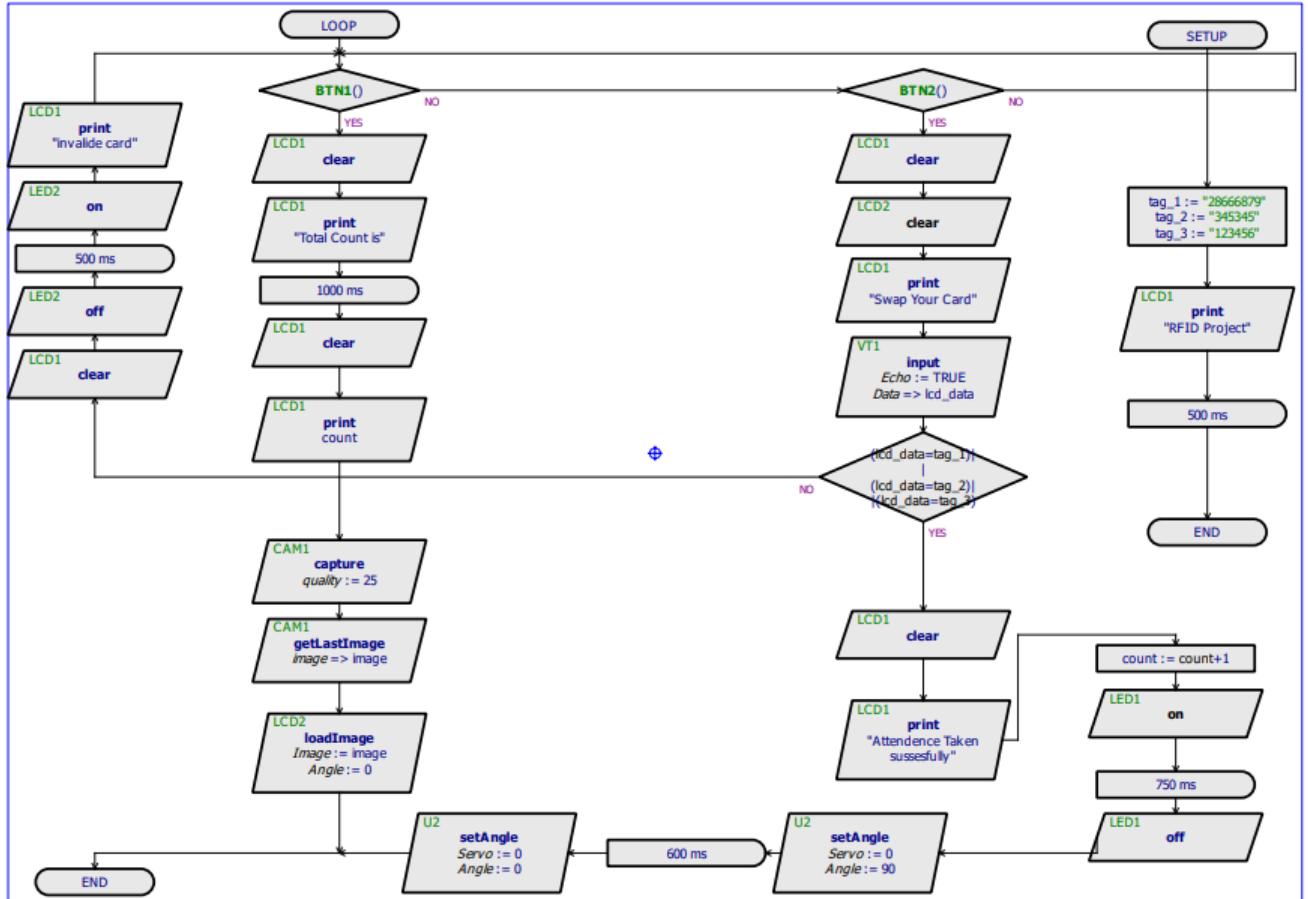


FIGURE 3.2 – Simulation des composants logiciels

3.1.4 Simulation des scénarios d'utilisation

Nous avons simulé les différents scénarios d'utilisation suivants : Utilisation de la carte RFID pour ouvrir la porte : Nous avons simulé la lecture de la carte RFID et la comparaison des informations de la carte avec la base de données. Nous avons vérifié que la porte s'ouvrait correctement si la carte RFID était reconnue et que la porte restait fermée si la carte RFID n'était pas reconnue.

Capture d’images de la caméra : Nous avons simulé la capture d’images de la caméra en utilisant le simulateur de caméra de Proteus. Nous avons vérifié que les images étaient capturées correctement et qu’elles étaient envoyées au Raspberry Pi. Nous avons également simulé des cas d’erreur, tels que la détection de cartes RFID non valides.

Nous avons vérifié que les signaux étaient correctement transmis entre les différents composants.

posants électroniques en utilisant l'outil d'analyse de signaux de proteus. Nous avons également vérifié que le code source était fonctionnel en simulant l'exécution du code dans l'environnement de simulation.

Les résultats de notre simulation ont montré que notre projet de porte intelligente était capable de détecter les cartes RFID valides et d'ouvrir la porte de manière précise avec le servo-moteur. Nous avons également vérifié que la caméra était capable de capturer des images avec succès.

Scenario 1 :carte valide

Si la carte RFID est valide la diode vert s'allume et le moteur se tourne pour l'ouverture de porte et un message de succès s'affiche sur l'afficheur lcd .

La figure 3.4 illustre ce scenario.

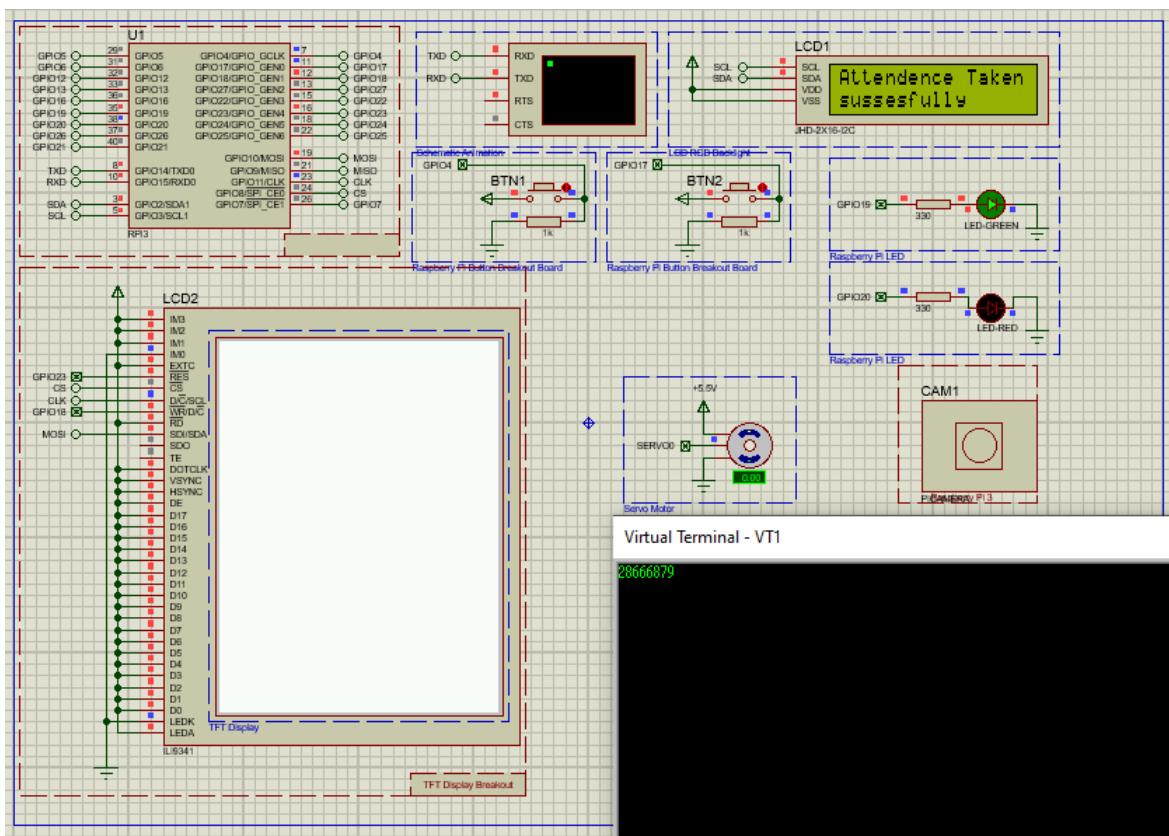


FIGURE 3.3 – Scenario 1 :carte valide

Scenario 2 :carte valide

Si la carte RFID n'est pas valide une led rouge s'allume et la camera capture une image et l'envoie par mail au proprietaire . et un message d'échec s'affiche sur l'afficheur lcd .

La figure 3.5 illustre ce scenario.

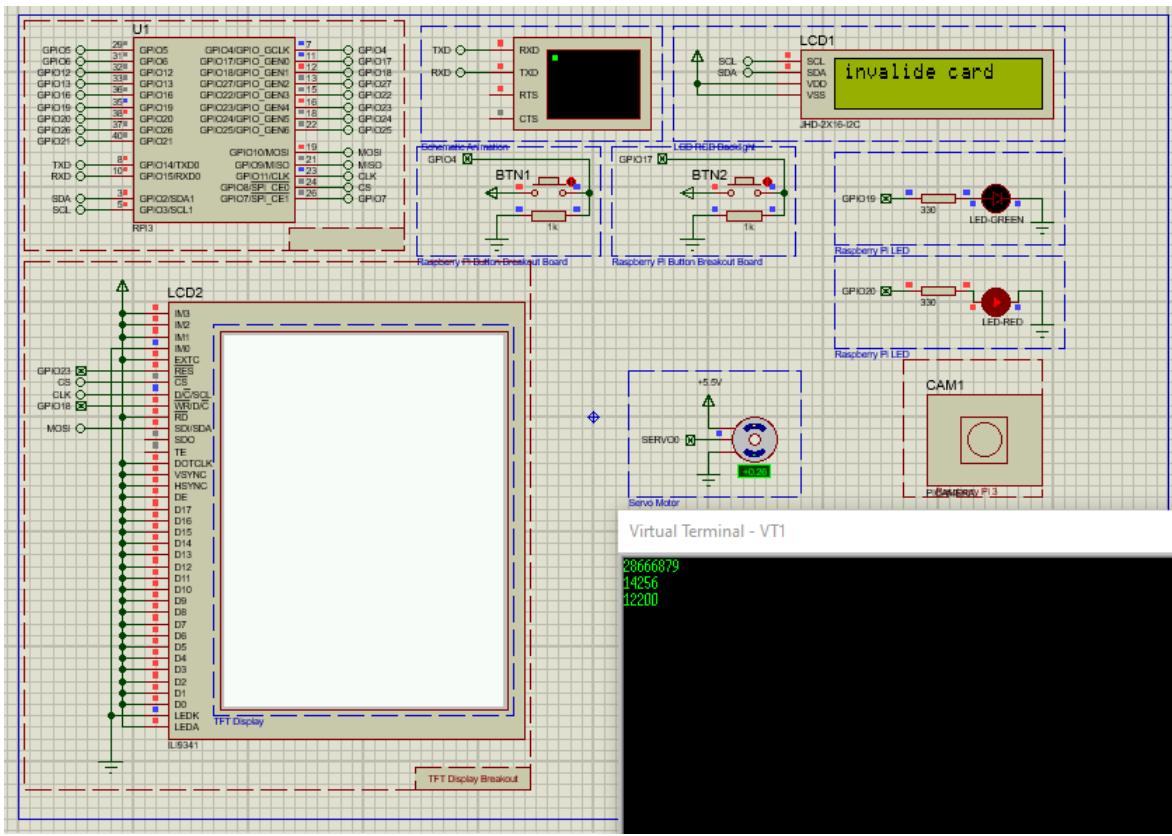


FIGURE 3.4 – Scenario 2 :carte valide

3.2 Limitations de la simulation

Malgré les nombreux avantages de la simulation dans Proteus, il y a des limites à ce qui peut être simulé. L'un des aspects qui ne peuvent pas être simulés dans Proteus est la réaction du système en temps réel à une situation donnée.

Par exemple, la rapidité de réponse d'un dispositif de sécurité en cas de détection d'un signal RFID ne peut être simulée que partiellement dans Proteus. Pour compenser ces limites, nous avons effectué des tests manuels en utilisant les composants réels pour valider les résultats de la simulation.

3.3 Perspectives d'amélioration de la simulation

Il y a plusieurs améliorations possibles pour la simulation de notre projet dans Proteus. Par exemple, nous pourrions ajouter des fonctionnalités supplémentaires pour simuler des situations plus complexes. Nous pourrions également utiliser des composants supplémentaires pour améliorer la précision de la simulation.

Enfin, nous pourrions ajuster les paramètres de simulation pour améliorer les résultats.

Par exemple, nous pourrions ajuster le temps de réponse de certains composants pour les rendre plus réalistes.

Conclusion

En conclusion, la simulation a été une étape importante dans la conception et la réalisation de notre projet de porte intelligente. Nous avons utilisé une approche de simulation pour valider notre conception et optimiser notre système. Les résultats de la simulation ont montré que notre système est capable de répondre aux exigences de fonctionnement et qu'il est fiable dans différentes situations. La simulation a été un outil précieux pour optimiser la conception de notre porte intelligente et pour nous aider à mieux comprendre son comportement.

CHAPITRE 4

RÉALISATION DU PROJET

Introduction	29
4.1 Environnement de travail :	29
4.1.1 Environnement matériel	29
4.2 montage physique	30
4.3 installation du logiciel :	31
4.4 Programmation :	32
4.5 test de systme :	33
4.6 Intégration et validation :	35
4.7 conception d'interface web :	37
Conclusion	41

Introduction

Ce chapitre est consacré à la description du processus de réalisation de notre système qui n'est autre que la mise en application de ce qui a été énoncé dans le chapitre précédent. Ceci en mettant en évidence l'ensemble des environnements (logiciels et matériels) de développement, de déploiement du système, ainsi qu'un aperçu sur les interfaces de l'application mobile et celle permettent la gestion du serveur.

Pratiquement notre réalisation est repartie sur trois volets : Le développement de l'interface du mail. La programmation de script serveur, La programmation et le câblage électronique du Raspberry.

4.1 Environnement de travail :

4.1.1 Environnement matériel

un ordinateur Pour la réalisation de notre projet, nous avons utilisé un ordinateur pour le développement de serveur et la configuration de raspberry . les Caractéristiques de machine utilisées :

	machine
Système d'exploitation	Windows 10
marque	asus
processeur	Intel core i7
Mémoire vive	8 Go ram

FIGURE 4.1 – les Caractéristiques de machine

Les terminaux mobiles :

Pour les opérations de test, dans le but d'avoir un rendu réel et la reception du mailcontenant des captures des images sur un appareil mobile nous avons utilisés un Smartphone dont les caractéristiques sont les suivantes :

	Smart phone
marque	iPhone
os	iOS
Mémoire vive	16 go

FIGURE 4.2 – les Caractéristiques Les terminaux mobiles :

4.2 montage physique

Raspberry Pi 4 Model B : c'est un mini-ordinateur avec une grande puissance de traitement qui nous permet d'exécuter notre programme et de contrôler les différents composants du système.

Module RFID : Nous avons utilisé un module RFID RC522 pour permettre l'identification des utilisateurs. Le module a été connecté au Raspberry Pi via une interface SPI.

caméra : Pour surveiller les personnes qui entrent ou sortent de la porte, nous avons installé une caméra Raspberry Pi Camera Module V2. La caméra a été connectée au Raspberry Pi via le port CSI.

diodes : Nous avons ajouté des diodes pour fournir des indications visuelles aux utilisateurs du système. Nous avons utilisé des diodes LED pour indiquer si l'accès a été accordé ou refusé.

afficheur lcd pour l'affichage des messages en cas de succès et d'échec d'accès.

Montage du circuit : Nous avons commencé par connecter le module RFID RC522 et les diodes LED au Raspberry Pi en suivant les schémas de connexion fournis par les fabricants. Nous avons également connecté la caméra Raspberry Pi Camera Module V2 au port CSI du Raspberry Pi.

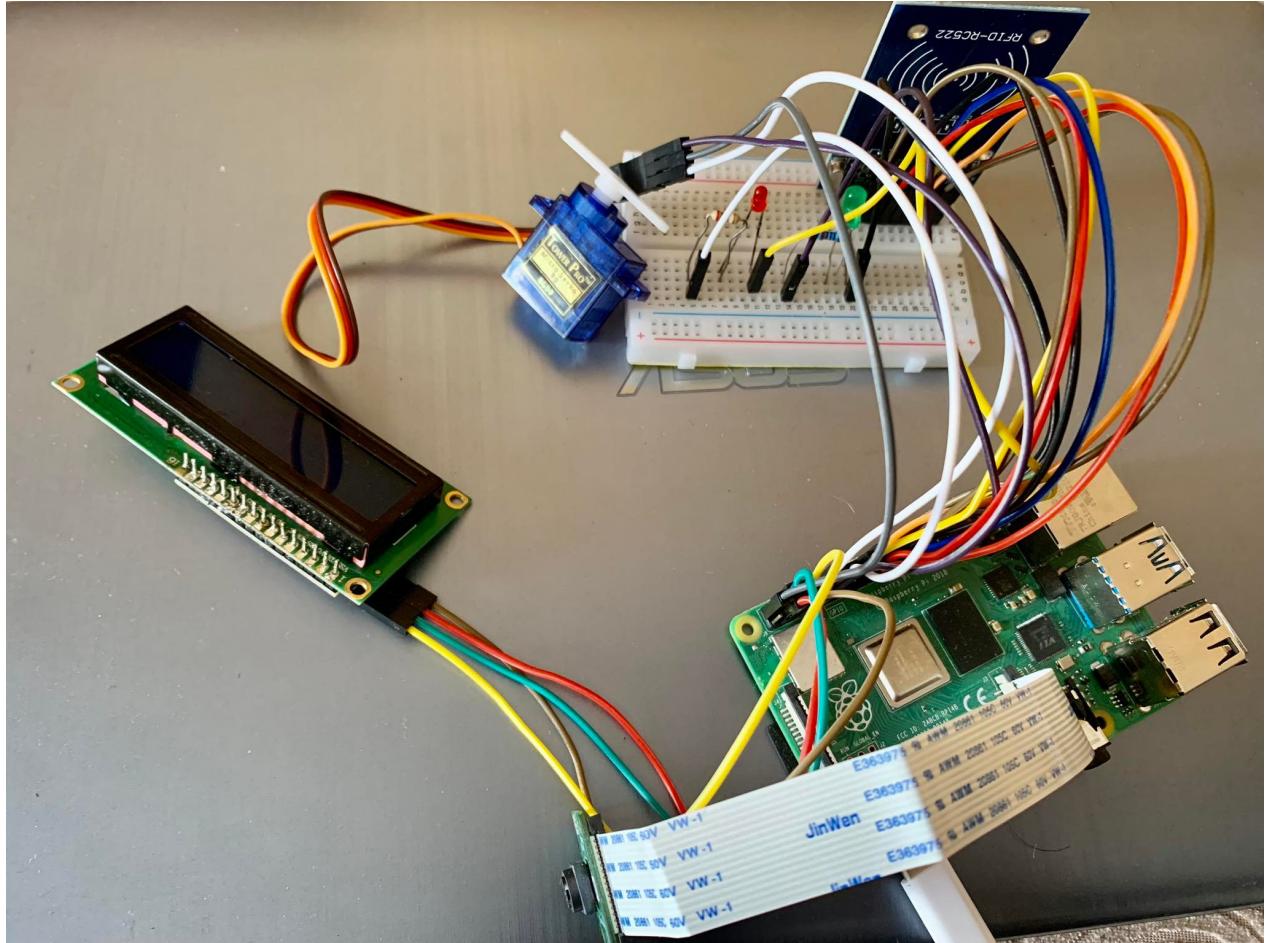


FIGURE 4.3 – montage physique

4.3 installation du logiciel :

Nous avons installé le système d'exploitation Raspbian sur le Raspberry Pi et avons configuré les paramètres réseau pour permettre la communication entre le Raspberry Pi et le module RFID. Nous avons également installé les bibliothèques Python nécessaires pour contrôler le module RFID, la caméra et les diodes LED.

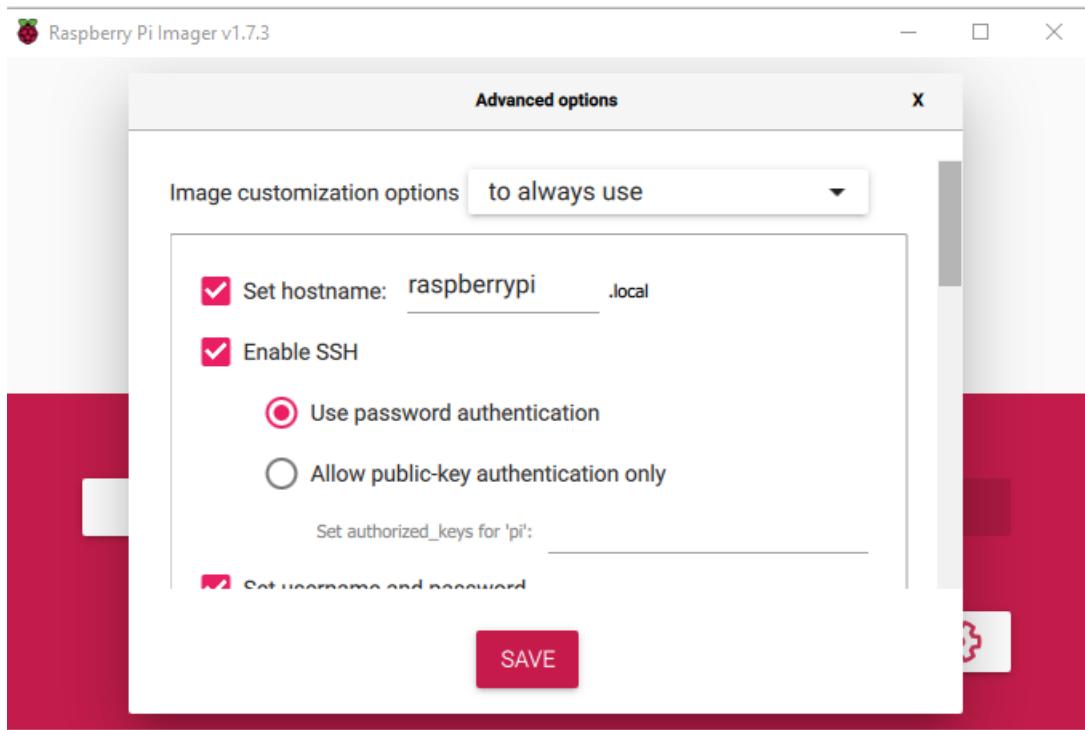


FIGURE 4.4 – montage physique

4.4 Programmation :

Dans cette partie nous allons fournir des détails sur la façon dont nous avons programmé le Raspberry Pi pour gérer les entrées et sorties de l'ensemble du système, y compris la manière dont nous avons configuré le module RFID pour reconnaître les cartes d'identité et comment nous avons utilisé la caméra pour prendre des photos des visiteurs. Nous avons écrit un programme Python qui utilise la bibliothèque MFRC522 pour lire les données d'identification des cartes RFID et les comparer aux données stockées dans la base de données. Si l'identification est valide, le programme allume la diode LED verte et envoie un signal pour déverrouiller la porte. Si l'identification est invalide, le programme allume la diode LED rouge et rejette l'accès.

FIGURE 4.5 – Programmation

4.5 test de systme :

Nous avons effectué plusieurs tests pour vérifier le bon fonctionnement de notre système. Nous avons testé l'identification des cartes RFID, la réaction des diodes LED et le contrôle de la caméra. Nous avons également testé l'accès à la porte pour nous assurer que le déverrouillage et le verrouillage étaient effectués correctement.

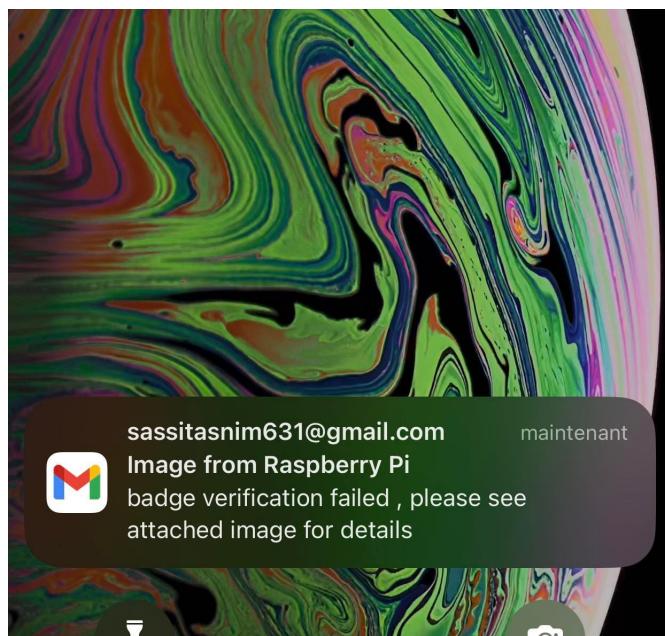


FIGURE 4.6 – envoi du mail

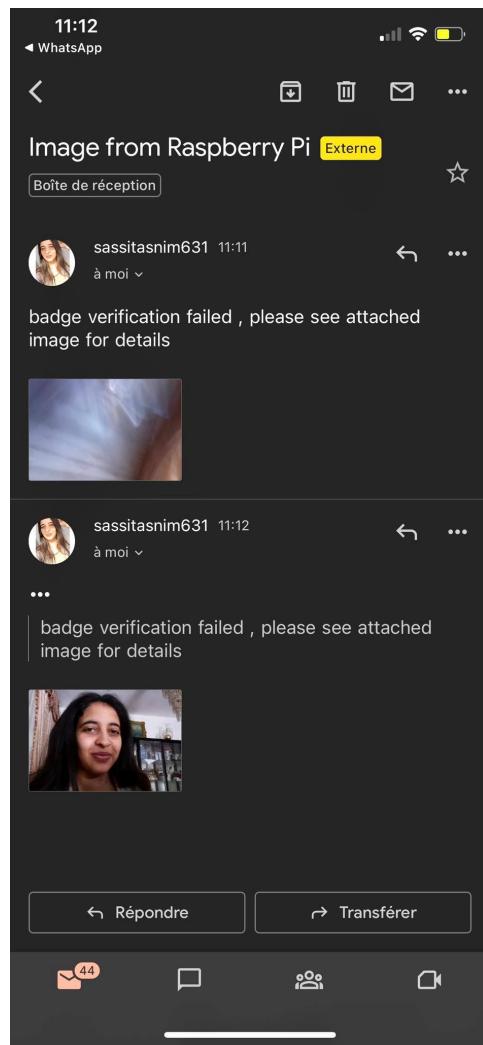


FIGURE 4.7 – envoi du mail

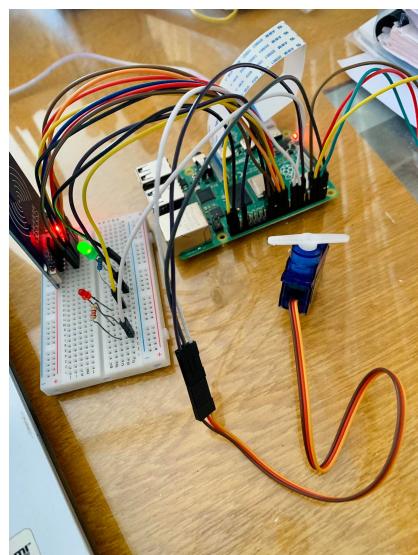


FIGURE 4.8 – test du carte valide

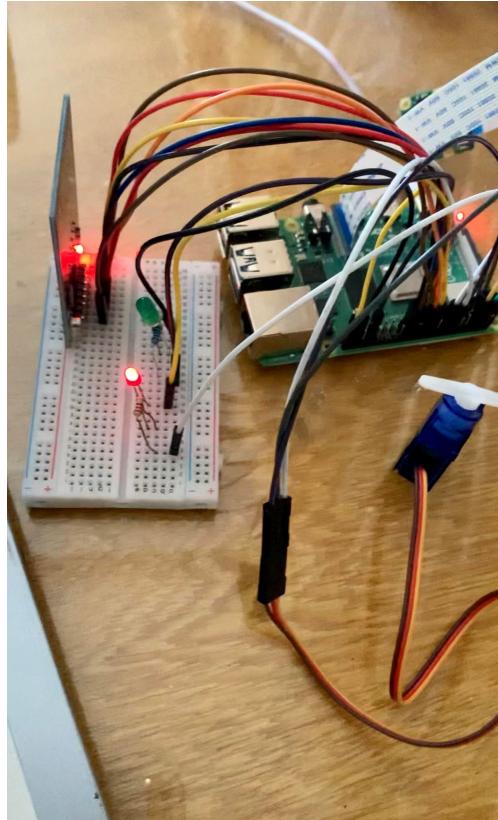


FIGURE 4.9 – test du carte invalide

4.6 Intégration et validation :

Nous avons intégré tous les composants du système et validé son bon fonctionnement en réalisant des tests de performance dans différentes conditions. Nous avons également vérifié la fiabilité et la sécurité du système.

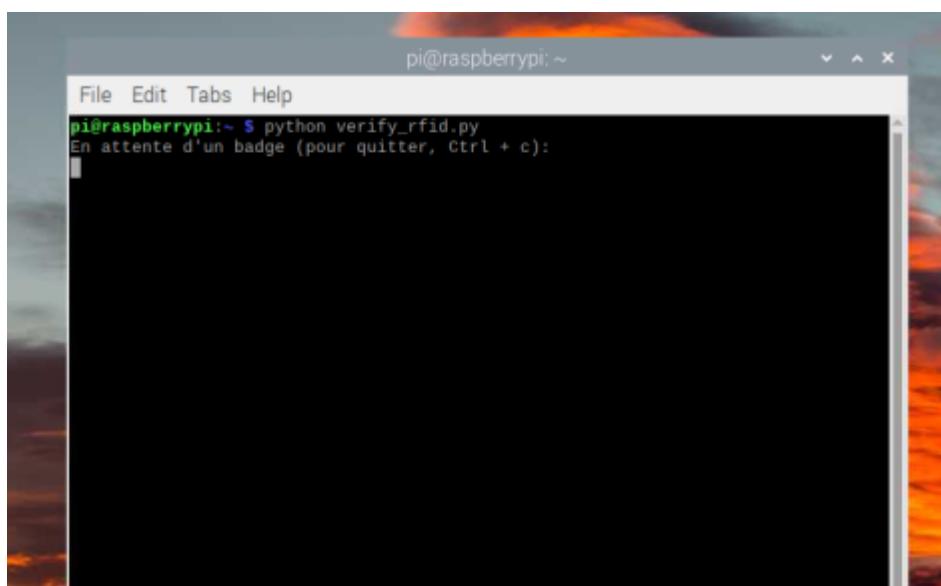


FIGURE 4.10 – attente du badge

```
pi@raspberrypi: ~
pi@raspberrypi: ~ login as: pi
pi@raspberrypi: ~ pi@192.168.137.51's password:
Linux raspberrypi 6.1.19-v8+ #1637 SMP PREEMPT Tue Mar 14 11:11:47 GMT 2023 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Mar 23 11:01:23 2023
pi@raspberrypi: ~ $ python verify_rfid.py
En attente d'un badge (pour quitter, Ctrl + c):
Badge [147, 112, 211, 27, 43] autorisé !
Badge [147, 112, 211, 27, 43] autorisé !
[ ]
```

FIGURE 4.11 – cas d'autorisation du carte

```
pi@raspberrypi: ~
pi@raspberrypi: ~ login as: pi
pi@raspberrypi: ~ pi@192.168.137.51's password:
Linux raspberrypi 6.1.19-v8+ #1637 SMP PREEMPT Tue Mar 14 11:11:47 GMT 2023 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Mar 23 11:01:23 2023
pi@raspberrypi: ~ $ python verify_rfid.py
En attente d'un badge (pour quitter, Ctrl + c):
Badge [147, 112, 211, 27, 43] autorisé !
Badge [147, 112, 211, 27, 43] autorisé !
[ ]
```

Windows taskbar showing open applications: WhatsApp - G..., Paramètres, VNC Viewer, 192.168.137.51 ..., Capture d'écran, pi@raspberrypi..., 17°C, FRA, 23/03/2023, 11:05.

FIGURE 4.12 – cas d'échec du carte

4.7 conception d'interface web :

Pour améliorer notre projet nous avons ajouté une interface web qui contrôle l'accès à distance des badges connectées à une base de données en ligne Firebase qui envoie des informations au raspberry pour l'ouverture du portail. La figure 4.13 suivante montre l'interface web.

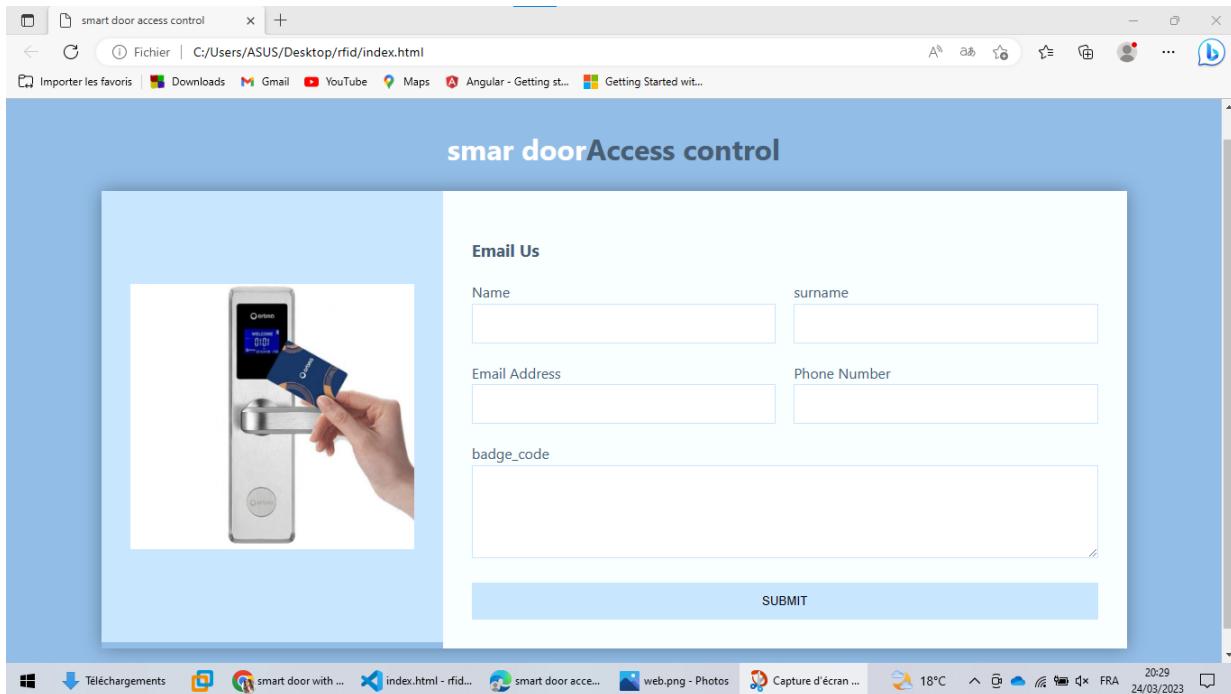


FIGURE 4.13 – interface web d'accès à distance

Firestore :

est un ensemble d'outils pour l'hébergement et le développement d'applications mobiles et web, qui permet l'envoi de notifications et de publicités, la remontée des erreurs et des clics effectués dans l'application.

nous avons créé une base de données en ligne nommée "rfid door" qui reçoit les données de l'interface web, faire une vérification et ensuite envoie les données au raspberry pour actionner .

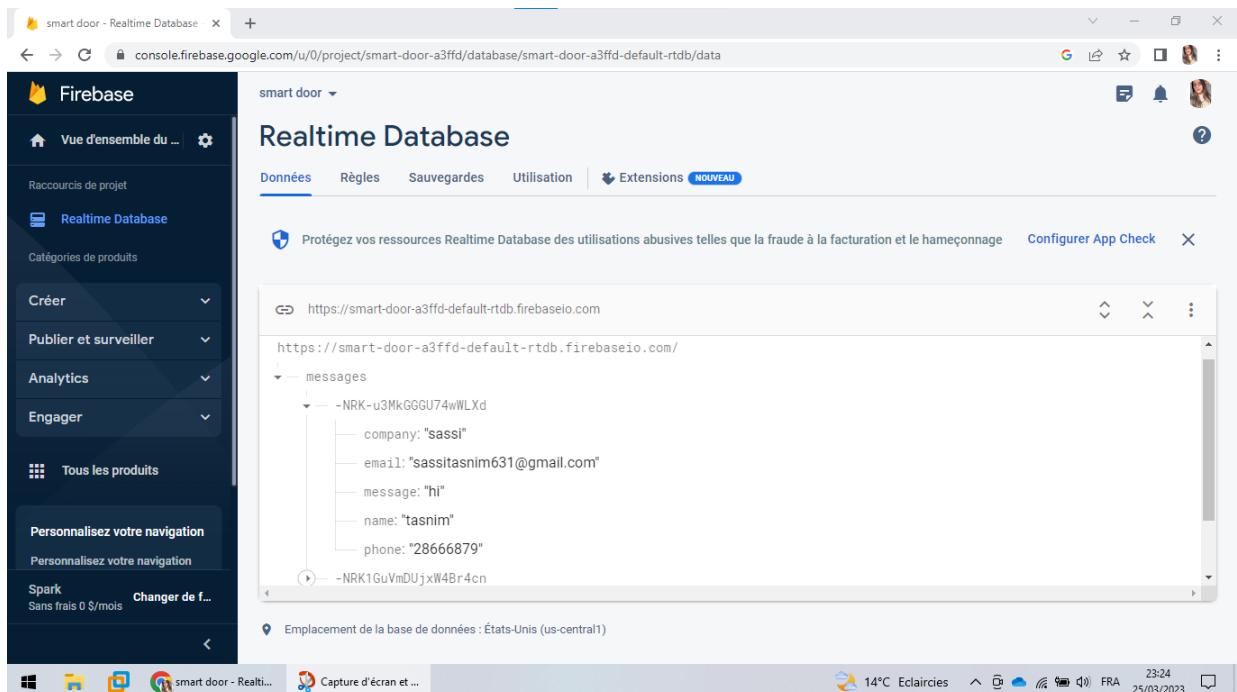


FIGURE 4.14 – Firebase

Connexion avec raspberry :

Si le badge entrée dans l'interface web est autorisé alors le moteur se tourne pour ouvrir la porte et la led verte s'allume avec un message de succès sinon la led rouge s'allume et un message d'erreur s'affiche.

The screenshot shows the Thonny IDE interface with two tabs: 'firebase.py' and 'verify_rfid.py'. The 'firebase.py' tab contains the following code:

```

1 import firebase_admin
2 from firebase_admin import credentials
3 from firebase_admin import db
4 import time
5 import RPi.GPIO as GPIO # Import the library to control GPIOs
6 import json
7 # Set up Firebase credentials and database reference
8 cred = credentials.Certificate('/home/pi/Documents/smart-door-a3ffd.firebaseio-admin.json')
9 firebase_admin.initialize_app(cred, {'databaseURL': 'https://smart-door-a3ffd.firebaseio'})
10 ref = db.reference('messages').order_by_child('timestamp').limit_to_last(1)
11
12 GPIO.setmode(GPIO.BOARD) # Set the pin numbering mode to Board
13 GPIO.setwarnings(False) # Disable warning messages
14
15 # Define the existing access code
16 existing_code = "123456"
17
18 LED_GREEN = 38 # Set the GPIO pin number for the green LED
19 LED_RED = 40 # Définit le numéro du port GPIO qui alimente la led rouge
20
21 # Define a function to turn on a LED
22 def turn_led_on(led):
23     GPIO.setup(led, GPIO.OUT) # Activate GPIO control
24     GPIO.output(led, GPIO.HIGH) # Turn on the LED
    
```

The terminal window on the right shows a command-line session:

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@student-fst.utm.tn: ~ 'name': 'sassitasnim631@gmail.com', 'phone': '28666879'
, 'surname': 'ss'))]
Traceback (most recent call last):
  File "/home/pi/firebase.py", line 88, in <module>
    ref = db.reference('messages').order_by_child('timestamp').start_at(list(messages.value())[0]['timestamp'] + 1).limit_to_last(1)
KeyError: 'timestamp'
pi@raspberrypi: ~ $ python firebase.py
Access denied
{
    "NR0hugCCBh12bHCU0h7": {
        "badge_code": "125",
        "email": "tasnim.sassie@student-fst.utm.tn",
        "name": "sassitasnim631@gmail.com",
        "phone": "28666879",
        "surname": "ss"
    }
}
Traceback (most recent call last):
  File "/home/pi/firebase.py", line 88, in <module>
    ref = db.reference('messages').order_by_child('timestamp').start_at(list(messages.value())[0]['timestamp'] + 1).limit_to_last(1)
KeyError: 'timestamp'
pi@raspberrypi: ~ $
    
```

FIGURE 4.15 – Connexion avec raspberry

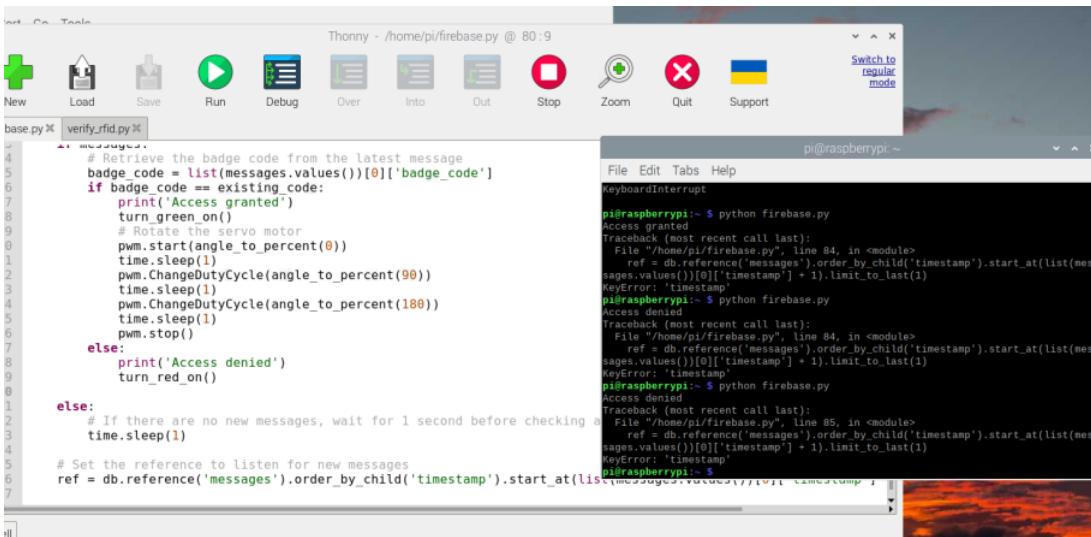


FIGURE 4.16 – Connexion avec raspberry

twilio :

Twilio est une plateforme de communication en ligne qui offre des API pour les développeurs permettant d'envoyer et de recevoir des messages textes, des appels vocaux et des vidéos, ainsi que d'utiliser des numéros de téléphone virtuels pour interagir avec les clients. L'API SMS de Twilio permet aux développeurs d'envoyer des messages SMS en utilisant le langage de programmation de leur choix.

Dans le cadre de notre projet, nous avons utilisé Twilio pour envoyer des messages SMS aux utilisateurs autorisés à accéder à notre système de porte intelligente. Nous avons pu créer un compte gratuit sur Twilio et obtenir un numéro de téléphone virtuel pour envoyer des messages.

Pour envoyer des SMS en utilisant l'API de Twilio, nous avons écrit un code Python qui utilisait la bibliothèque Twilio pour se connecter à l'API SMS de Twilio. Nous avons ensuite pu envoyer des messages SMS en utilisant le numéro de téléphone virtuel fourni par Twilio.

En résumé, Twilio s'est avéré être un outil puissant et facile à utiliser pour envoyer des messages SMS en utilisant un langage de programmation de notre choix. Grâce à l'API SMS de Twilio, nous avons pu ajouter une fonctionnalité de messagerie à notre système de porte intelligente, permettant ainsi aux utilisateurs autorisés de recevoir des notifications en temps réel sur leur téléphone portable.

The screenshot shows the Twilio Console homepage. At the top, it says "Console My first Twilio account" and "Trial: \$15.50 Upgrade". Below that, there are tabs for "Develop" and "Monitor". A sidebar on the left has links for "Phone Numbers", "Messaging", "Studio", and "Voice", along with "Explore Products" and "Docs and Support". The main content area features a welcome message: "Ahoy sassi, welcome to Twilio! Learn to build your first SMS app by following these steps." It includes four numbered steps: 1. Get a number, 2. Try out SMS, 3. Check out docs, and 4. Invite and upgrade. Step 1 is expanded with the title "Step 1. Get a Twilio phone number" and an icon of a phone connected to a cloud. Text explains that a virtual phone number is needed for sending or receiving SMS. A note says "While your account is in trial, you can get one free USA or Canadian phone number." Step 1 also includes a checked checkbox for "You've got a phone number!" and a link to "View it in Account info below".

FIGURE 4.17 – twilio

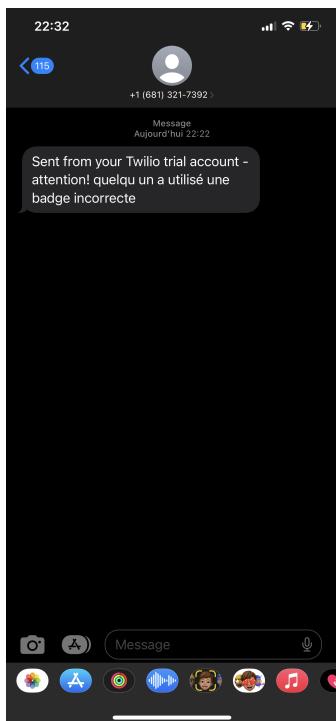


FIGURE 4.18 – Envoi d'un SMS

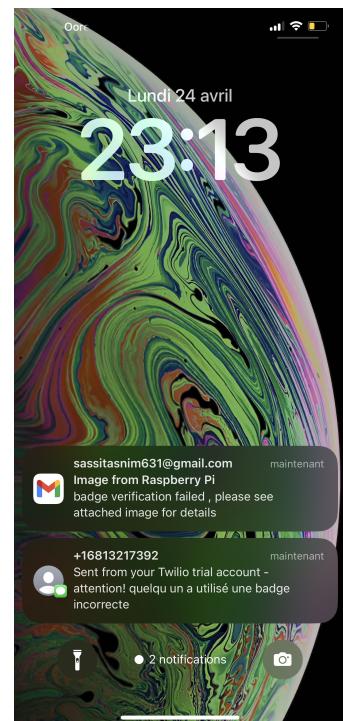


FIGURE 4.19 – Envoi d'un mail

```
pi@raspberrypi:~ $ python verify_rfid.py
En attente d'un badge (pour quitter, Ctrl + c):
Badge [147, 112, 211, 27, 43] autorisé !
Badge [227, 12, 137, 46, 72] interdit !
Message sent to +21628666879 with ID: SMd2lc176d49991dcf8cb39d06de80b93f
Image captured.
```

FIGURE 4.20 – verification

Conclusion

Dans ce dernier chapitre nous avons présenté ce système avec ses différentes parties et leur principe de fonctionnement ainsi que sa réalisation pratique. En effet, Grâce aux étapes citées dans ce chapitre, nous avons réussi à réaliser un système de Smart Door avec RFID fonctionnel, fiable et sécurisé en utilisant le Raspberry Pi, le module RFID, la caméra et les diodes LED.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Afin d'implémenter notre porte intelligente, une étude des systèmes y impliqués nous a permis de comprendre de près les technologies et les composants concernés par le domaine de l'intelligence appliquée à l'habitat. Elle nous a introduits à l'environnement du Raspberry et à la conception et la mise en service de serveur d'application.

La réalisation de notre système a été initiée par l'étude des solutions existantes et leur critique afin de fixer les fonctions du futur système. Ensuite, nous avons établi une analyse détaillée de système projeté modélisée en langage sysml et powerAMC . Enfin, nous avons concrétisé notre travail par la construction de prototype de notre solution.

Notre étude traite le concept de l'intelligence dans un espace réduit celui de la maison. Faisons intervenir un horizon plus vaste, nous introduisons à la notion de l'environnement intelligent et proprement dit l'intelligence ambiante. Il s'agit de doter un environnement (espaces commerciaux, hôpitaux, sociétés) de capacité de perception, d'analyse, de traitement et de prise de décision. Donc elle doit impérativement intervenir plus de domaines, de technologie et des composants.

Ce travail illustre le fonctionnement d'un système de verrouillage électronique intelligent à base d'un Raspberry Pi , Les aspects de ce système sont : la détection du badge RFID à l'aide d'un lecteur RFID, puis vérification d'un premier code personnel à l'aide d'une recherche dans une première table en deuxième étape, puis en une troisième l'autorisation d'accès si le badge est correcte en actionnant le moteur sinon capturer l'image et l'envoyer par mail vers le propriétaire de la maison ou société .

Nous avons rencontré quelques obstacles dans ce travail, tels que l'existence d'un défaut de

certains composants, le manque des instruments dans les laboratoires, d'où la nécessité de les chercher ailleurs, le flux d'Internet faible au niveau des laboratoires, le langage de programmes que nous n'avions pas connus auparavant. Mais nous avons tiré parti de nouvelles informations et avons travaillé avec de nombreux programmes (Python, , Raspbian ...). Et nous nous sommes familiarisés avec les différents types de ce système de verrouillage. Cependant, ce projet peut être amélioré, nous pouvons ajouter plus de paramètres à l'utilisateur pour lui donner plus de contrôle, et lui permettre aussi de contrôler les accès à partir d'un réseau internet externe, ainsi de commander le verrouillage et le déverrouillage à distance.