



---

# RAPPORT TP

**PROJET COMMUNICATION SANS FIL**

---



Rediger: MAMBO MOTSOU JUNIOR DEOGRACIAS

## Table of Contents

INTRODUCTION.....	3
I. Principe de fonctionnement de la carte .....	4
II. Cahier des charges.....	5
1. Architecture de la carte .....	6
a. Identification .....	6
b. Compteur.....	7
II. Connect et Disconnect.....	9
A. Connect.....	9
B. Disconnect.....	9
III. Sélection de la carte.....	10
IV. Mise à jour des informations dans la carte .....	11
V. Incrémentation et décrémentation .....	12
VI. Difficultés Rencontrées et Leçons Apprises dans le Projet sur la Carte MIFARE Classic .....	15
Difficultés Rencontrées .....	15
CONCLUSION .....	17

Figure 1:Schema Fonctionnel.....	5
Figure 2:Configuration du Secteur2 de la memoire .....	6
Figure 3: Configuration du Secteur3 de la memoire .....	7
Figure 4: Interface de l'application.....	8

# INTRODUCTION

Ce rapport expose le processus de conception d'une application dédiée à la technologie de carte à puce MIFARE Classic, conformément au cahier des charges fourni par notre enseignant. Les cartes MIFARE Classic, basées sur la norme ISO/IEC 14443 Type A, offrent des possibilités d'identification sécurisée sans contact, avec des applications variées dans la sécurité, le transport, et la gestion des accès. Guidés par le cahier des charges, nous avons entrepris une démarche méthodique, de la recherche à l'implémentation, afin de créer une application répondant aux exigences définies. Ce rapport détaille les étapes du processus de conception, les défis rencontrés, les solutions adoptées, et les perspectives d'amélioration, illustrant ainsi notre engagement à appliquer nos connaissances tout en contribuant au progrès des solutions technologiques dans le domaine des cartes à puce.

## **I. Principe de fonctionnement de la carte**

Le principe de fonctionnement de la carte MIFARE Classic repose sur la technologie de communication sans contact RFID (Radio-Frequency Identification). Cette carte, conforme à la norme ISO/IEC 14443 Type A, utilise une puce intégrée qui communique avec un lecteur RFID via des ondes radio. La carte MIFARE Classic est dotée d'une antenne et d'une mémoire EEPROM (Electrically Erasable Programmable Read-Only Memory). Lorsqu'elle entre en proximité d'un lecteur RFID, l'antenne de la carte capte l'énergie émise par le champ radio du lecteur, activant ainsi la puce. La communication s'établit alors entre le lecteur et la carte, permettant l'échange d'informations.

La mémoire EEPROM de la carte MIFARE Classic est divisée en secteurs, chacun étant protégé par des clés d'authentification. Les données stockées dans ces secteurs peuvent être lues ou écrites après une procédure d'authentification réussie. Ainsi, la carte peut être utilisée pour stocker des informations sensibles telles que des identifiants, des clés d'accès, ou d'autres données spécifiques à l'application.

Lorsqu'un utilisateur présente sa carte au lecteur, celui-ci émet un signal pour interagir avec la carte. Après une authentification réussie, les informations stockées sur la carte peuvent être utilisées pour permettre l'accès à des zones sécurisées, activer des services ou effectuer des transactions, selon les applications spécifiques mises en œuvre. La carte MIFARE Classic, en raison de sa facilité d'utilisation et de sa sécurité relative, a trouvé des applications étendues dans divers domaines, notamment le contrôle d'accès, les transports en commun et les systèmes de paiement.

## II. Cahier des charges

L'interface graphique doit pouvoir :

Écrire et lire l'identité de la personne dans la carte (nom et prénom) ;

Incrémenter et décrémenter des unités dans la carte (avec la gestion d'un backup) ;

Une information visuelle et/ou sonore du coupleur doit être faite à chaque transaction.

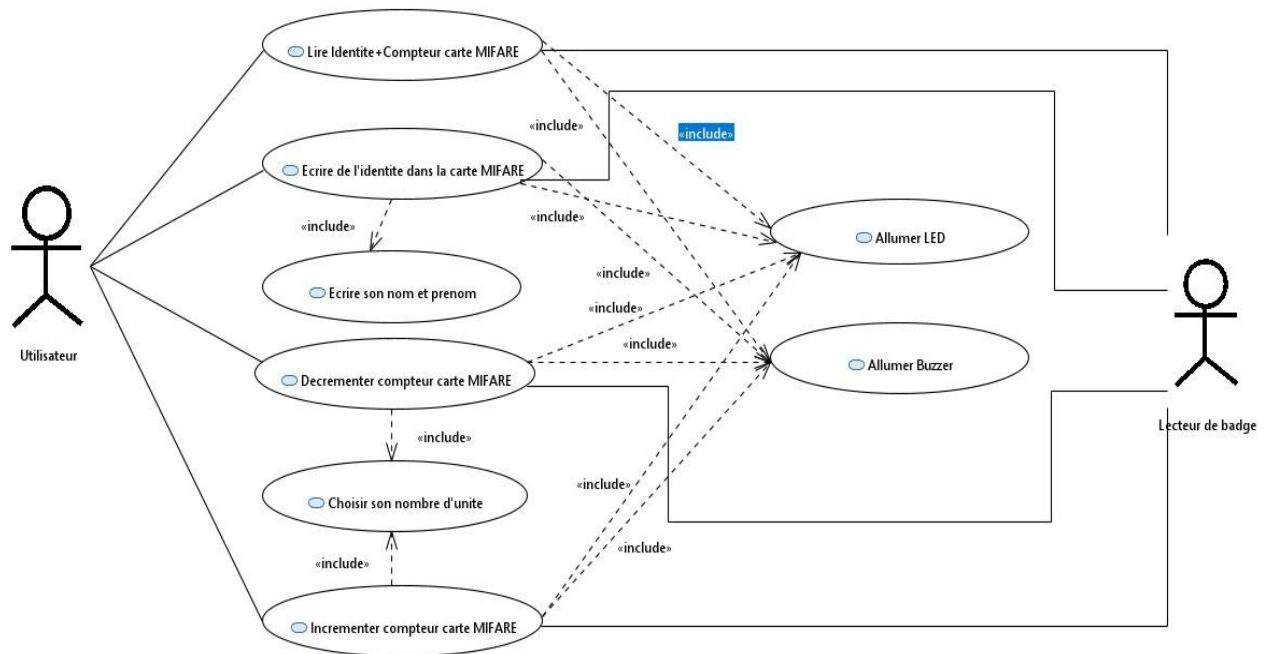


Figure 1:Schema Fonctionnel

Les différentes fonctionnalités de l'interface graphique sera défini dans les paragraphe suivant

## 1. Architecture de la carte

Afin que les cartes soient lues par toutes les applications. Nous définirons une architecture commune à chaque carte.

### a. Identification

Pour l'identification, la carte aura cette configuration: 2 blocks de données + 1 block contenant le nom de l'application « Identité » .

Sector	Block	Data
2	11	KeyA+AccessBit+KeyB
	10	Nom
	9	Prenom
	8	« Identite »

Figure 2: Configuration du Secteur2 de la memoire

L'authentification se fera avec les clés suivantes:

- Pour la lecture nous utiliserons **la clé A** situé dans le Secure Element du lecteur (coffre fort) à l'index 2
- Pour l'écriture nous utiliserons **la clé B** situé dans le Secure Element du lecteur (coffre fort) à l'index 2

## b. Compteur

Pour le compteur, la carte aura cette configuration: 2 blocks de compteurs, 1 block contenant le nom de l'application «Porte-monnaie » .

L'authentification se fera avec les clés suivantes :

- Pour la lecture et la décrémentation nous utiliserons la clé A situé dans le Secure Element du lecteur (coffre-fort) à l'index 3
- Pour l'écriture et l'incrémentation nous utiliserons la clé B situé dans le Secure Element du lecteur (coffre-fort) à l'index 3

Sector	Block	Data
3	15	KeyA+AccessBit+KeyB
	14	Compteur
	13	Backup Compteur
	12	« Porte Monnaie »

Figure 3: Configuration du Secteur3 de la memoire



### III. Réalisation de l'application

Une documentation regroupant les différentes fonctions utilisables avec le lecteur et la carte nous a été fournie par l'enseignant. C'est en utilisant cette documentation que nous avons réalisé notre interface. Dans cette partie de notre rapport nous allons présenter les étapes de développement tout en expliquant les différents fonctionnements.

Nous avons commencé par réaliser une interface sur qt, qui au début ne faisait aucune communication. Au terme de nos séances de TP, nous avons obtenu le rendu suivant :

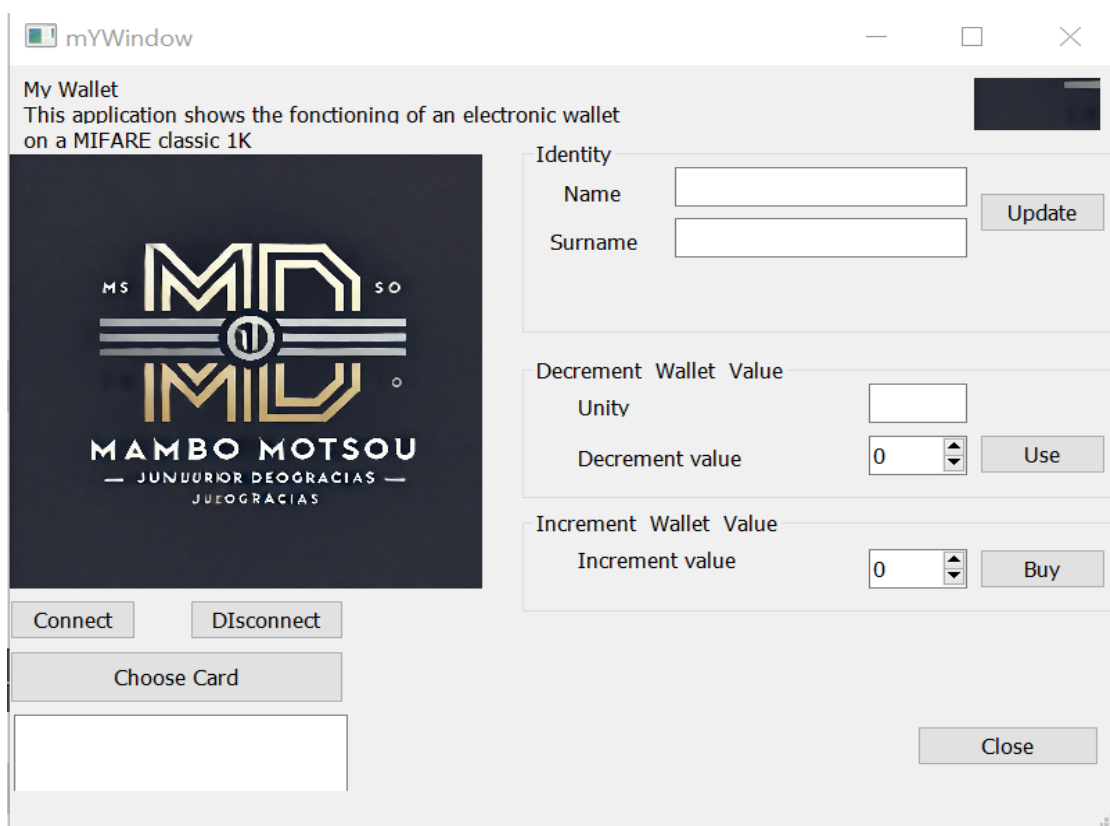


Figure 4: Interface de l'application

Après avoir réalisé cette interface, nous avons utilisé qt en passant par des slots afin de commencer à communiquer .

## II. Connect et Disconnect

### A. Connect

La fonction void **on\_Connect\_clicked()** est déclenchée lorsqu'on clique sur le bouton connect. Une variable status de type int16\_t est déclarée et initialisée à la valeur MI\_OK. MI\_OK est utilisé pour représenter un état de réussite. La fonction OpenCOM est appelée avec la structure MonLecteur en tant que paramètre pour ouvrir la communication avec le lecteur de carte. Le statut résultant est stocké dans la variable status et affiché dans la console via qDebug. La fonction **Version** est appelée pour obtenir la version du lecteur. La version est ensuite affichée dans un élément d'interface utilisateur (un widget texte) appelé **Affichage**. La fonction LEDBuzzer est appelée pour contrôler les LED du lecteur. Dans ce cas, la LED rouge est activée (LED\_RED\_ON) et la LED verte est éteinte (LED\_GREEN\_OFF).

### B. Disconnect

Lorsque l'utilisateur clique sur le bouton disconnect la fonction **on\_disconnect\_clicked()** est déclenchée. Dans cette fonction, la fonction CloseCOM est appelée avec la structure MonLecteur en tant que paramètre pour fermer la communication avec le lecteur de carte. Le statut résultant est stocké dans la variable status.

Lorsqu'il clique sur le bouton **close**, la fonction **on\_Close\_clicked()** est déclenchée. Dans cette fonction, la fonction quit() de l'application Qt est appelée pour fermer l'application. Cela met fin au programme en cours d'exécution.

### III. Sélection de la carte

Après avoir fini de configurer les options de connexion et déconnexion, nous sommes passés à la détection de la carte. Lorsqu'on clique sur le bouton « Choose Carte » C'est la fonction `on_Select_clicked()` qui est déclenchée et elle fonctionne comme suit :

- Déclaration de variables

```
uint16_t status = MI_OK;  
uint8_t atq[2]; uint8_t  
sak[1]; uint8_t uid[12];  
uint16_t uidlen = 12;  
uint32_t pValue = 0;
```

Plusieurs variables sont déclarées pour stocker des informations telles que le statut des opérations, les données de la carte (ATQ, SAK, UID), la longueur de l'UID et une valeur ('pValue') que nous prenons dans la carte et qui correspond au nombre d'unités.

- Polling pour détecter la carte MIFARE Classic

```
status = ISO14443_3_A_PollCard(&MonLecteur, atq, sak, uid, &uidlen);
```

La fonction « `ISO14443_3_A_PollCard` » est utilisée pour détecter la présence d'une carte MIFARE Classic. Les informations sur la carte (ATQ, SAK, UID) sont récupérées.

- Vérification de la présence d'une carte

```
if (status == MI_OK) {
```

Cette condition vérifie si la détection de la carte a été réussie.

- Contrôle des LEDs du lecteur

```
LEDBuzzer(&MonLecteur, LED_GREEN_ON+LED_YELLOW_ON+LED_RED_ON+  
LED_GREEN_ON);
```

```
DELAYS_MS(10);
```

```
LEDBuzzer(&MonLecteur, LED_GREEN_ON + LED_YELLOW_ON);
```

Les LEDs du lecteur sont manipulées pour indiquer visuellement la détection de la carte.

- Lecture du prénom depuis un bloc de la carte :

```
status = Mf_Classic_Read_Block(&MonLecteur, true, 9, data, true, 2);
```

La fonction « `Mf_Classic_Read_Block` » est utilisée pour lire les données d'un bloc spécifique (bloc 9) de la carte MIFARE Classic. Les données sont ensuite converties en une chaîne de caractères (prénom) et affichées dans une interface utilisateur.

- Lecture du nom depuis un autre bloc de la carte :

De manière similaire à la lecture du prénom, le nom est lu depuis un autre bloc (bloc 10) de la carte, converti en chaîne de caractères et affiché dans l'interface utilisateur.

- Lecture du nombre d'unités la carte :

```
status = Mf_Classic_Read_Value(&MonLecteur, TRUE, 14, &pValue, AuthKeyA, 3 );  
ui->Nombre_Unite->setText(QString::number(pValue));
```

La fonction « Mf\_Classic\_Read\_Value » est utilisée pour lire le nombre d'unités depuis un secteur spécifique (secteur 14) de la carte MIFARE Classic. La valeur lue est affichée dans un élément d'interface utilisateur.

#### **IV. Mise à jour des informations dans la carte**

Après avoir sélectionné la carte, on peut modifier le nom et le prénom dans carte en modifiant les champs de texte correspondant dans l'interface. Puis il suffit de cliquer sur le bouton mise à jour et les informations dans la carte sont modifiées. Lorsque l'on clique sur ce bouton, c'est la fonction **void MaFenetre::on\_Mise\_a\_jour\_clicked()** qui est déclenchée. Celle-ci fonctionne comme suit :

- Déclaration de variables:

```
uint8_t atq[2];  
uint8_t sak[1];  
uint8_t uid[12];  
uint16_t uid_len = 12;
```

Des variables sont déclarées pour stocker les informations de la carte (ATQ, SAK, UID) lors de la détection de la carte MIFARE Classic.

- Contrôle des LEDs du lecteur :

```
LEDBuzzer(&MonLecteur,  
LED_GREEN_ON+LED_YELLOW_ON+LED_RED_ON+LED_GREEN_ON);  
DELAYS_MS(1); LEDBuzzer(&MonLecteur, LED_GREEN_ON);
```

Les LEDs du lecteur sont manipulées pour indiquer visuellement un certain état, pour signaler à l'utilisateur que la carte est en cours de traitement.

- Détection de la carte MIFARE Classic :

**ISO14443\_3\_A\_PollCard(&MonLecteur, atq, sak, uid, &uid\_len);**

La fonction « ISO14443\_3\_A\_PollCard » est utilisée pour détecter la présence d'une carte MIFARE Classic et récupérer les informations de la carte.

- Ecriture sur la carte:

**char DataIn1[16];**

**strncpy(DataIn1, ui->Nom->toPlainText().toUtf8().data(), 16);**

**Mf\_Classic\_Write\_Block(&MonLecteur, TRUE, 10, (uint8\_t\*)DataIn1, AuthKeyB, 2 );**

Le nom est extrait de l'élément d'interface utilisateur « ui->Nom », converti en une chaîne de caractères et écrit dans un bloc spécifique (bloc 10) de la carte MIFARE Classic.

- Écriture sur la carte:

**char DataIn[16];**

**strncpy(DataIn, ui->Prenom->toPlainText().toUtf8().data(), 16);**

**Mf\_Classic\_Write\_Block(&MonLecteur,TRUE,9,(uint8\_t\*)DataIn, AuthKeyB,2);**

Le prénom est extrait de l'élément d'interface utilisateur « ui->Prenom », converti en une chaîne de caractères et écrit dans un autre bloc (bloc 9) de la carte MIFARE Classic.

## **v. Incrémentation et décrémentation**

Pour incrémenter des unités dans la carte, l'utilisateur utilise le bouton « charger » en précisant le nombre d'unités qu'il veut et pour décrémentation il utilisera le bouton « payer ». Lorsqu'il clique sur le bouton charger, c'est la fonction **on\_charger\_clicked()** qui est déclenchée et elle fonctionne comme suit :

- Déclaration de variables:

**uint32\_t pValue = 0;**

**uint8\_t atq[2];**

**uint8\_t sak[1];**

**uint8\_t uid[12];**

**uint16\_t uid\_len = 12;**

**uint32\_t ValeurEcrire = 0;**

Des variables sont déclarées pour stocker des informations telles que le solde actuel de la carte (`pValue`), les données de la carte (ATQ, SAK, UID), la longueur de l'UID, et la valeur à incrémenter (`ValeurEcrire`).

- Lecture de la carte MIFARE Classic :

**ISO14443\_3\_A\_PollCard(&MonLecteur, atq, sak, uid, &uid\_len);**

La fonction « **ISO14443\_3\_A\_PollCard** » est utilisée pour détecter et lire les informations de la carte MIFARE Classic

- Incrémentation de la valeur sur la carte:

**Mf\_Classic\_Increment\_Value(&MonLecteur, TRUE, 14, Valeur\_Increment, 13, AuthKeyB, 3);**

La fonction « **Mf\_Classic\_Increment\_Value** » est utilisée pour incrémenter la valeur stockée dans le secteur 14 de la carte, en utilisant la clé d'authentification « **AuthKeyB** ».

- Lecture de la nouvelle valeur sur la carte:

**Mf\_Classic\_Read\_Value(&MonLecteur, TRUE, 14, &pValue, AuthKeyA, 3 );**

**ui->Nombre\_Unite->setText(QString::number(pValue));**

La fonction « **Mf\_Classic\_Read\_Value** » est utilisée pour lire la nouvelle valeur du secteur 14 de la carte après l'incrémentation. Cette valeur est ensuite affichée dans un élément d'interface utilisateur.

- Contrôle des LEDs du lecteur :

**LEDBuzzer(&MonLecteur,  
LED\_GREEN\_ON+LED\_YELLOW\_ON+LED\_RED\_ON+LED\_GREEN\_ON);  
DELAYS\_MS(500); LEDBuzzer(&MonLecteur, LED\_RED\_ON);**

Le lecteur est manipulé pour indiquer le statut de l'opération de chargement à travers un son.

Lorsque l'utilisateur va cliquer sur le bouton payer, c'est la fonction **on\_payer\_clicked()** qui est déclenchée et elle fonctionne comme suit:

- Déclaration de variables:

**uint32\_t Valeur\_Decrement = 0;**

Une variable « **Valeurdecrementer** » est déclarée pour stocker la valeur à décrémenter.

- Lecture de la valeur à décrémenter depuis l'interface utilisateur:

**Valeurdecrementer = ui ->decrementer\_unite->value();**

La valeur à décrémenter est lue depuis un élément d'interface utilisateur.

- Décrémentation de la valeur sur la carte

**Mf\_Classic\_Decrement\_Value(&MonLecteur, TRUE, 14, Valeurdecrementer, 13, AuthKeyA, 3 );**

La fonction « **Mf\_Classic\_Decrement\_Value** » est utilisée pour décrémenter la valeur stockée dans le secteur 14 de la carte, en utilisant la clé d'authentification « **AuthKeyA** ».

La suite de la fonction est la même que celle décrite précédemment.

## **vi. Difficultés Rencontrées et Leçons Apprises dans le Projet sur la Carte MIFARE Classic**

### **Difficultés Rencontrées**

Travailler sur le projet de la carte MIFARE Classic a été une expérience enrichissante, mais non sans défis. L'un des obstacles les plus persistants a été l'état défectueux de la barre d'espace de mon ordinateur. Ce problème matériel, bien que mineur en apparence, a considérablement affecté mon efficacité, m'obligeant à trouver des solutions créatives pour coder et déboguer correctement malgré cette contrainte.

Un autre défi majeur a été la compréhension des subtilités du protocole MIFARE Classic. Cette technologie exige une solide maîtrise des protocoles de communication bas niveau, comme le RFID et le NFC, ainsi que des méthodes de chiffrement spécifiques, notamment Crypto1. L'équilibre entre l'apprentissage théorique et la mise en œuvre pratique a représenté une courbe d'apprentissage exigeante, en particulier lors du débogage des comportements inattendus dans le processus de communication entre la carte et le lecteur.

De plus, intégrer ce projet dans le cadre du développement avec Qt a présenté son propre lot de difficultés. Comprendre et utiliser l'architecture événementielle de Qt était nouveau pour moi. J'ai dû assurer une interaction fluide entre l'interface utilisateur et les processus de back-end responsables de l'authentification et de la lecture/écriture des données sur la carte.

### **Leçons Apprises**

Malgré ces obstacles, ce projet s'est révélé extrêmement enrichissant et m'a permis d'apprendre plusieurs leçons précieuses :

#### **1. Résilience face aux Problèmes :**

- La barre d'espace défectueuse m'a appris la patience et la créativité dans la résolution des problèmes techniques. J'ai compris que même des problèmes matériels mineurs peuvent impacter la productivité, mais que la persévérance permet de les surmonter.

#### **2. Compréhension Approfondie de la Technologie MIFARE Classic :**

- J'ai acquis une connaissance détaillée du fonctionnement des cartes MIFARE Classic, notamment leur structure en blocs, leurs méthodes d'authentification et leurs mécanismes de stockage de données. Apprendre à interfacer la carte avec le matériel m'a donné une expérience précieuse dans les systèmes embarqués et les protocoles de communication.

#### **3. Maîtrise du Framework Qt :**

- Travailler avec Qt m'a permis de renforcer mes compétences en programmation, notamment dans la conception d'interfaces utilisateur interactives et la gestion des communications signal-slot. J'ai appris à intégrer des fonctionnalités matérielles dans un cadre logiciel, rendant l'application à la fois robuste et conviviale.



#### **4. Importance du Débogage et des Tests :**

- Résoudre les problèmes de communication avec la carte m'a montré l'importance de tester méthodiquement chaque étape. J'ai appris à décomposer des problèmes complexes en tâches plus simples et gérables pour les résoudre efficacement.

#### **5. Adaptabilité et Ingéniosité :**

- De la recherche sur le chiffrement Crypto1 au débogage des problèmes de communication RFID, ce projet m'a appris à apprendre rapidement et à appliquer mes connaissances théoriques à des défis pratiques.

### **Conclusion**

Ce projet n'a pas seulement consisté à concevoir un système fonctionnel pour interagir avec une carte MIFARE Classic. Il a représenté un parcours personnel de croissance et de découverte. Surmonter des difficultés techniques, comme une barre d'espace défaillante et des protocoles complexes, m'a permis de développer une meilleure capacité d'adaptation et de résolution de problèmes. Ces apprentissages seront sans aucun doute des atouts précieux pour mes projets futurs et ma carrière professionnelle.

## CONCLUSION

En conclusion, ce projet a abouti au développement réussi d'une application fonctionnant avec une carte MIFARE Classic, offrant des fonctionnalités de lecture, écriture, décrémentation et incrémentation, tout en assurant une gestion efficace des éléments matériels tels que les LEDs et le buzzer du lecteur. L'implémentation de mécanismes de sécurité, tels que l'authentification avec des clés spécifiques, renforce la protection des données. Ce travail a permis d'acquérir des compétences essentielles dans la manipulation des cartes MIFARE, la communication avec le matériel embarqué, et la création d'une interface graphique. En définitive, ce projet représente une étape significative dans l'exploration des applications basées sur les cartes MIFARE Classic, offrant une base solide pour des développements futurs et démontrant la capacité à concevoir des solutions interactives et sécurisées dans le domaine des technologies sans contact.