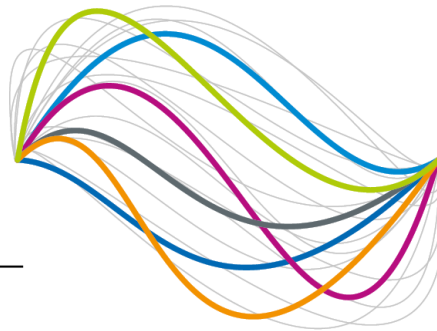


UBFC

UNIVERSITÉ  
BOURGOGNE FRANCHE-COMTÉ



## Communication sans Fil



## Table des matières

Carte Mifare sans contact .....	3
Quelques caractéristiques sur la carte .....	3
Détails du programme .....	4
1 <sup>ère</sup> Etape : .....	4
2 <sup>ème</sup> étape : .....	5
3 <sup>ème</sup> étape : .....	6
Conclusion .....	7

## Carte Mifare sans contact

Le tp se déroulera en deux parties :

- Une première partie sur l'étude de la documentation du lecteur et de la carte
- Une seconde partie sur la mise en œuvre des fonctions apprises lors de la première partie et l'implémentation d'un programme fonctionnel permettant de lire et d'écrire sur une carte mifare classic 1k à l'aide de Qt creator

## Quelques caractéristiques sur la carte

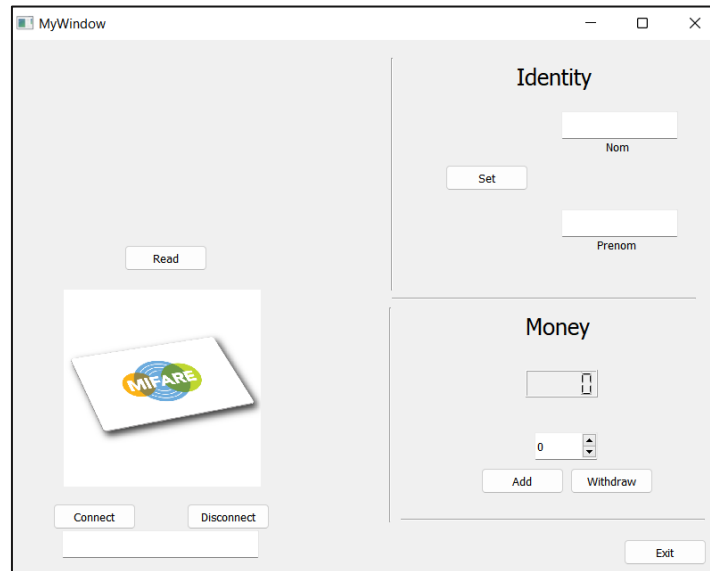
Mifare Classic est un type de carte à puce sans contact largement utilisé pour le contrôle d'accès, la billetterie et d'autres applications.

La carte Mifare Classic a une capacité de mémoire de 1KB ou 4KB, selon la version. Elle utilise un algorithme de cryptage propriétaire appelé Crypto-1 pour la sécurité. Toutefois, cet algorithme s'est révélé vulnérable à certains types d'attaques, ce qui permet aux pirates d'accéder aux données de la carte.

L'un des principaux avantages de la carte Mifare Classic est sa compatibilité avec un large éventail de lecteurs et de systèmes. Elle est également largement disponible et relativement peu coûteuse, ce qui en fait un choix populaire pour de nombreuses applications.

## Détails du programme

Nous écrivons un programme capable de se connecter au lecteur de carte, ainsi que d'écrire et de lire les informations sur celle-ci, à savoir le nom, le prénom, le montant inscrit sur la carte. Voici un aperçu de l'interface :



Pour exploiter toutes les fonctions de la carte, nous allons dans un premier temps inclure les bibliothèques ODALID sur notre projet QT.

Le fonctionnement de l'interface a été simplifiée pour une ergonomie optimale afin que les utilisateurs puissent s'en servir facilement.

### 1<sup>ère</sup> Etape :

Sur la gauche de l'interface, il y a deux boutons « connect » et « disconnect » pour se connecter au coupleur. La version du coupleur s'affiche dans la « textbox » juste en dessous. Il est important de noter qu'il faut toujours déconnecter le coupleur avant de le reconnecter, car si non le programme ne fonctionne plus. En effet, deux liaisons séries seront ouvertes simultanément pour un seul coupleur, ce qui corrompra les données reçues.

La connexion et la déconnexion sont possible grâce à ses deux fonctions :

```
status=OpenCOM(&lecteur);
```

Figure 2 : Connexion au coupleur

```
status=CloseCOM(&lecteur);
```

Figure 1: Déconnexion du coupleur

De plus, pour récupérer et afficher la version du coupleur utilisée, on inclut :

```
status=Version(&lecteur);
```

Figure 3 : Récupération de la version du coupleur

2<sup>ème</sup> étape :

Une fois le coupleur connecté, il suffit de disposer la carte sur le lecteur et de cliquer sur le bouton « Read ». Les informations contenues sur la carte apparaissent sur l'interface dans les cases « Nom », « Prénom » et « Money ». Pour avoir accès à ces informations, nous avons besoin de chercher dans les différents blocs de stockage de la carte, comme indiqué sur les figures 4 et 5 :

Sect or	Blo ck	Data
2	11	KeyA+AccessBit+KeyB
	10	Nom
	9	Prenom
	8	« Indentite »

Figure 4: chemin pour accéder à l'identité inscrite sur la carte

Sect or	Blo ck	Data
3	15	KeyA+AccessBit+KeyB
	14	Compteur
	13	Backup Compteur
	12	« Porte Monnaie »

Figure 5: Chemin pour accéder au compte du porte-monnaie

Ainsi, on récupère le nom avec cette fonction :

```
status = Mf_Classic_Read_Block(&lecteur, TRUE, 10, data, AuthKeyA, 2);
```

Figure 6: Lecture du nom enregistré sur la carte (Block10, secteur 2)

On récupère le prénom avec la même fonction, mais cette fois si en lisant le Block 9 :

```
status = Mf_Classic_Read_Block(&lecteur, TRUE, 9, data, AuthKeyA, 2);
```

Figure 7: lecture du Prénom enregistré sur la carte

On utilise la fonction « Read\_value » pour lire le solde inscrit sur la carte, mais avec le secteur 3 et le Block 14 :

```
status=Mf_Classic_Read_Value(&lecteur, TRUE, 14, &data2, AuthKeyA, 3);
```

Figure 8 : lecture du montant inscrit dans la carte

3<sup>ème</sup> étape :

Nous allons à présent voir comment inscrire de nouvelles informations sur la carte :

Pour écrire une nouvelle identité : il faut écrire tout simplement dans les « textbox » nom et prénom la nouvelle identité et appuyer sur le bouton « set » pour enregistrer sur la carte.

Pour cela, on récupère tout simplement le nom et prénom marqué dans les textbox et on les écrits dans la mémoire correspondante de la carte, à savoir le secteur 2 block 10 pour le nom et secteur 9 pour le prénom.

```
strncpy(nom, ui->Nom->toPlainText().toUtf8().data(), 16);  
status = Mf_Classic_Write_Block(&lecteur, TRUE, 10, (uint8_t*) nom, AuthKeyB, 2);
```

Figure 9: enregistrement du nouveau nom

```
strncpy(prenom, ui->Prenom->toPlainText().toUtf8().data(), 16);  
status = Mf_Classic_Write_Block(&lecteur, TRUE, 9, (uint8_t*) prenom, AuthKeyB, 2);
```

Figure 10: enregistrement du nouveau prénom

Pour gérer le portefeuille, il suffit de rentrer un nombre dans le « spinbox » et d'appuyer sur « add » ou « withdraw » pour ajouter ou retirer de la somme actuelle. Pour se faire, nous ajoutons (ou retirons) la valeur de la « spin box » au block 14 de la carte (soit le backup), et nous copions ce backup que nous mettons dans le block 13 de la carte correspondant au compteur :

```
status=Mf_Classic_Increment_Value(&lecteur, TRUE, 14, ui->spin_Money->value(), 13, AuthKeyB, 3);  
status=Mf_Classic_Restore_Value(&lecteur, TRUE, 13, 14, AuthKeyA, 3); //on prend la valeur dans
```

Figure 11: incrémentation de l'argent sur la carte

La nouvelle valeur est ensuite automatiquement affichée sur l'interface

## Conclusion

Il nous a été permis durant ce TP de nous familiariser avec le matériel de la société Odalid, connue pour son implication dans la modernisation des moyens de transactions billettiques au sein des transports publics notamment. Le fait de créer une interface graphique pour se servir de la carte nous met dans les conditions pour apprendre à se servir des différentes fonctions qui lui sont associés, chercher dans la doc et se servir des librairies incluent avec.