

Compte rendu de TP de Communication Sans Fil

Carte sans-contact MIFARE Classic

Lucas CHABIRON / Lucas_Chabiron@etu.u-bourgogne.fr
Aubin SEPTIER / Aubin_Septier@etu.u-bourgogne.fr

Sommaire

Introduction.....	4
Fonctionnement de la carte sans-contact MIFARE Classic.....	4
Fonctionnement de notre programme.....	4
Fonctionnement de notre interface.....	6
Conclusion.....	9

Introduction

Dans ce projet de Communication sans fil, l'objectif était de réaliser l'interface graphique permettant la gestion d'une carte sans-contact MIFARE Classic.

Nous avons utilisé pour cela l'IDE [Qt](#) afin de réaliser notre projet et l'interface, ainsi que [Github](#) pour héberger et versionner notre projet.

Le code source de notre projet est disponible sur [Github](#).

Nous avons utilisé la documentation d'[Odalid](#) afin de pouvoir appliquer les fonctions à notre disposition et de manipuler la carte sans-contact et le lecteur.

Dans ce compte-rendu, nous allons présenter comment fonctionne notre programme, les aspects techniques liés à la carte sans-contact MIFARE Classic ainsi que le fonctionnement de notre interface.

Fonctionnement de la carte sans-contact MIFARE Classic

Dans un premier temps, nous devons étudier la carte sans-contact MIFARE Classic afin de comprendre son fonctionnement.

La carte MIFARE Classic fonctionne sur le principe de la technologie RFID, qui permet une communication sans contact entre la carte et le lecteur. Elle est alimentée par les ondes radio émises par le lecteur lorsqu'elle entre dans son champ, ce qui active la puce intégrée à la carte. Dotée d'une mémoire segmentée en secteurs et blocs, la MIFARE Classic possède des mécanismes de sécurité sous forme de clés A et B pour chaque secteur, assurant la protection des données contre les accès non autorisés. Pour lire ou écrire des données, le lecteur doit s'authentifier auprès de la carte avec la clé correspondante, permettant ainsi l'accès sécurisé aux blocs de données. Cette combinaison de communication sans fil rapide, de stockage sécurisé et de mécanismes d'authentification rend la carte MIFARE Classic particulièrement adaptée à des applications variées telles que les systèmes de paiement rapide, les contrôles d'accès ou les systèmes de billetterie, tout en offrant la durabilité et la réutilisabilité nécessaires pour des usages répétés.

Dans la structure de la carte MIFARE Classic détaillée dans la documentation fournie, nous observons une organisation en secteurs et blocs, chacun attribué à une fonction spécifique. Le secteur 2 est consacré à l'identification avec des blocs alloués au stockage du nom et du prénom de l'utilisateur, ainsi qu'un bloc réservé pour les clés de sécurisation et les bits d'accès, essentiels pour la protection des données personnelles. L'accès à ce secteur est régulé par l'utilisation de clés différentes pour la lecture et l'écriture, assurant ainsi que les informations d'identification ne peuvent être lues ou modifiées que par des acteurs authentifiés. Parallèlement, le secteur 3 sert de compteur, avec un bloc dédié à la

sauvegarde de la valeur du compteur et un autre à son opérationnel, permettant des transactions ou le suivi d'une valeur cumulative. Ici aussi, l'accès est finement contrôlé par des clés qui permettent soit la lecture et la décrémentation, soit l'écriture et l'incrémentation de la valeur du compteur. Cette séparation fonctionnelle entre identification et comptage, renforcée par des mesures de sécurité, reflète l'efficacité de la conception de la carte pour répondre aux exigences spécifiques de transactions sécurisées et de gestion d'identité.

Fonctionnement de notre programme

Notre programme est découpé en plusieurs dossiers et fichiers. Le dossier "LIB" contient la librairie fournie nous permettant de manipuler et d'interagir avec la carte sans-contact MIFARE Classic. Le second dossier, intitulé "assets", est celui où l'on stocke les images de notre interface. Notre fichier ".gitignore" nous assure de ne pas mettre de fichiers indésirables dans notre repository Github. Le fichier "Mifare_Wallet.pro" contient les informations de notre projet Qt. Nous avons ensuite les fichiers "main.cpp", "mainwindow.cpp" et "mainwindow.h" sur lesquels nous reviendrons juste après. Le fichier "mainwindow.ui" est le fichier contenant les informations de l'interface graphique et donc des éléments, ou widgets, qui seront manipulables dans les trois fichiers '.cpp' et '.h' mentionnés précédemment. Enfin, le fichier "resources.qrc" permet de gérer les images du dossier "assets" avec un chemin relatif.

Le fichier "main.cpp" permet uniquement de lancer l'application lors de l'exécution de celle-ci. Le fichier "mainwindow.h" est celui où l'on déclare nos fonctions que nous utiliserons dans le fichier qui nous intéresse le plus, le fichier "mainwindow.cpp".

Le fichier "mainwindow.cpp" est le fichier contenant les fonctions nous permettant d'activer les fonctionnalités de l'application. Nous allons donc étudier son contenu en détail.

Nous avons tout d'abord des fichiers d'en-tête nécessaires à notre projet qui sont inclus. Ces fichiers contiennent les déclarations de toutes les classes, fonctions et variables que nous utiliserons dans notre code.

Nous avons ensuite le constructeur de notre classe qui sera appelé lors du lancement de notre projet. L'expression "ui(new Ui::MainWindow)" initialisera notre interface utilisateur.

Nous avons ensuite deux lignes, la première dont le but sera de créer et initialiser les widgets et layouts définis dans le fichier '.ui' et la seconde qui va définir le texte indiquant que nous n'avons pas de lecteur de carte connecté.

Nous avons ensuite inséré deux images dans l'interface, défini un destructeur de classe et une variable globale "MyReader" qui sera utilisée pour accéder au lecteur de carte.

La première fonction, "on_Connect_clicked()" s'active lorsque le bouton 'Connect' de l'interface est cliqué. Elle ouvre alors la communication avec le lecteur de carte, récupère la version du lecteur et l'affiche dans l'encadré souhaité. Puis la LED jaune du lecteur s'allume.

La seconde fonction, `on_Disconnect_clicked()` s'active lorsque le bouton "Disconnect" est cliqué, comme dans le cas précédent. Elle éteint les LED du lecteur, ferme la communication avec le lecteur et met à jour l'affichage afin d'indiquer qu'aucun lecteur n'est connecté.

La fonction `on_Quit_clicked()` éteint les LED du lecteur, ferme la communication avec le lecteur et ferme l'application lorsque le bouton "Quit" est activé.

En appuyant sur le bouton "ConnectCard", on active la fonction `on_ConnectCard_clicked()` qui va allumer le buzzer sonore du lecteur et essayer de détecter une carte sur le lecteur et de lire les informations de celle-ci afin de mettre à jour l'interface utilisateur. Si la carte n'est pas trouvée, alors une erreur est affichée sur l'interface, sinon la LED verte s'allume sur le lecteur et les informations de la carte sont affichées à l'utilisateur.

La fonction `on_Update_clicked()` nous permet de récupérer les informations contenues dans les cases souhaitées (celles du nom et du prénom), limitées à 16 caractères. Nous utilisons la fonction `Mf_Classic_Write_Block()` de la librairie fournie afin de pouvoir écrire dans les blocs 9 et 10, comme indiqué dans la partie sur le fonctionnement de la carte MIFARE Classic. La fonction permet ainsi de modifier l'identité de l'utilisateur de la carte et met à jour l'affichage en conséquence.

Les fonctions `on_Pay_clicked()` et `on_Load_clicked()` permettent de décrémenter et incrémenter le compteur de la carte sans-contact.

La première fonction, utilisée pour décrémenter, vérifie d'abord que la valeur à décrémenter n'est pas supérieure au solde total. Si ce n'est pas le cas, alors un message d'erreur s'affiche et l'opération échoue. Sinon, le buzzer du lecteur est déclenché et le compteur est décrémenté de la valeur souhaitée. La fonction `Mf_Classic_Decrement_Value()` est utilisée sur le bloc 13 contenant la valeur du compteur afin de décrémenter ce dernier. L'affichage utilisateur est ensuite mis à jour afin d'afficher la nouvelle valeur du compteur.

La seconde fonction sert à incrémenter le compteur. Elle utilise la fonction `Mf_Classic_Increment_Value()` afin d'augmenter la valeur du compteur, soit la valeur du bloc 13 de la carte sans-contact. Le buzzer du lecteur se déclenche à nouveau et la valeur affichée est mise à jour pour l'utilisateur.

Finalement, la dernière fonction est `readCard()`. Cette fonction permet de lire les informations de la carte à l'aide du lecteur. Elle lit à la fois les informations d'identité, c'est-à-dire le nom et le prénom, et le solde du compteur de la carte. Elle utilise la fonction `Mf_Classic_Read_Sector()` afin de pouvoir lire le secteur souhaité. Ici, la fonction va lire le secteur 2 pour récupérer les informations relatives à l'identité. Le nom et le prénom récupérés doivent ensuite être convertis de l'hexadécimal vers des caractères afin d'être lisible pour l'utilisateur. Ensuite, la fonction va récupérer la valeur du compteur à l'aide de `Mf_Classic_Read_Value()` appliqué sur le bloc 13 contenant la donnée voulue. Dans les deux cas, si la lecture échoue, une erreur est envoyée à l'utilisateur. Si la lecture réussit, alors les informations sont affichées sur l'interface utilisateur.

Chacune des fonctions renvoie un message erreur à l'utilisateur en cas d'erreurs afin d'informer l'utilisateur du problème.

Lors de la programmation, la partie la plus “complexe” était de bien lire les bons blocs et secteurs et de bien comprendre comment utiliser les fonctions fournies par la librairie externe. De plus, nous devons faire attention à bien convertir les données de la carte pour les rendre lisibles à l'utilisateur et inversement pour écrire sur la carte.

Fonctionnement de notre interface

L'interface de notre projet est basée sur celle présentée en exemple dans l'énoncé du TP de Communication Sans Fil.

Nous pouvons la visualiser sur l'image ci-dessous :

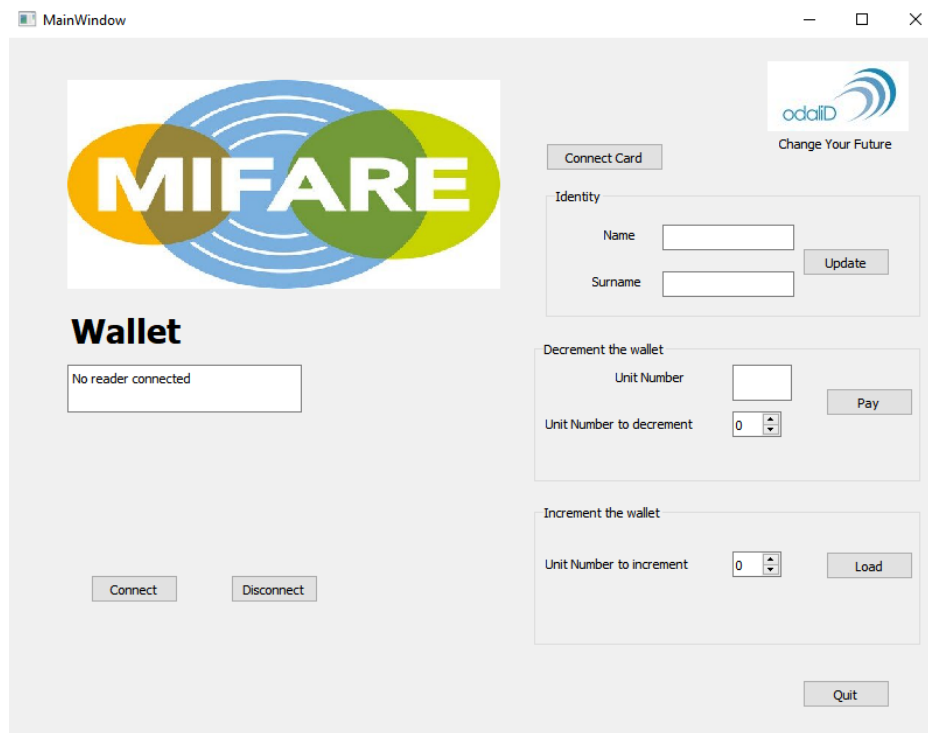


Figure 1 : Interface de notre application de gestion de carte MIFARE Classic

Notre application est divisée en plusieurs sous-parties. Nous avons, en bas à gauche, deux boutons “Connect” et “Disconnect” permettant de détecter et de connecter ou de déconnecter le lecteur de carte branché par USB. Lors de la connexion, le nom du lecteur s'affiche alors dans l'encadré prévu à cet effet (là où est écrit “No reader connected” sur notre image ci-dessus) et la LED jaune de l'appareil s'allume. Au moment de la déconnexion, la LED allumée s'éteint et le lecteur est déconnecté de l'application. Le nom affiché dans l'encadré se change afin d'afficher de nouveau “No reader connected” afin d'informer l'utilisateur que le lecteur n'est plus connecté.

Une fois le lecteur détecté et connecté, nous pouvons lire notre carte à l'aide de celui-ci en plaçant notre carte sur le lecteur et en appuyant sur le bouton "Connect Card", en haut à droite. Une fois la carte détectée et lue, les informations de celle-ci, c'est-à-dire l'identité et la valeur du compteur, s'afficheront en-dessous dans les zones destinées à cet usage. La LED verte s'allume alors afin d'indiquer la bonne lecture de la carte.

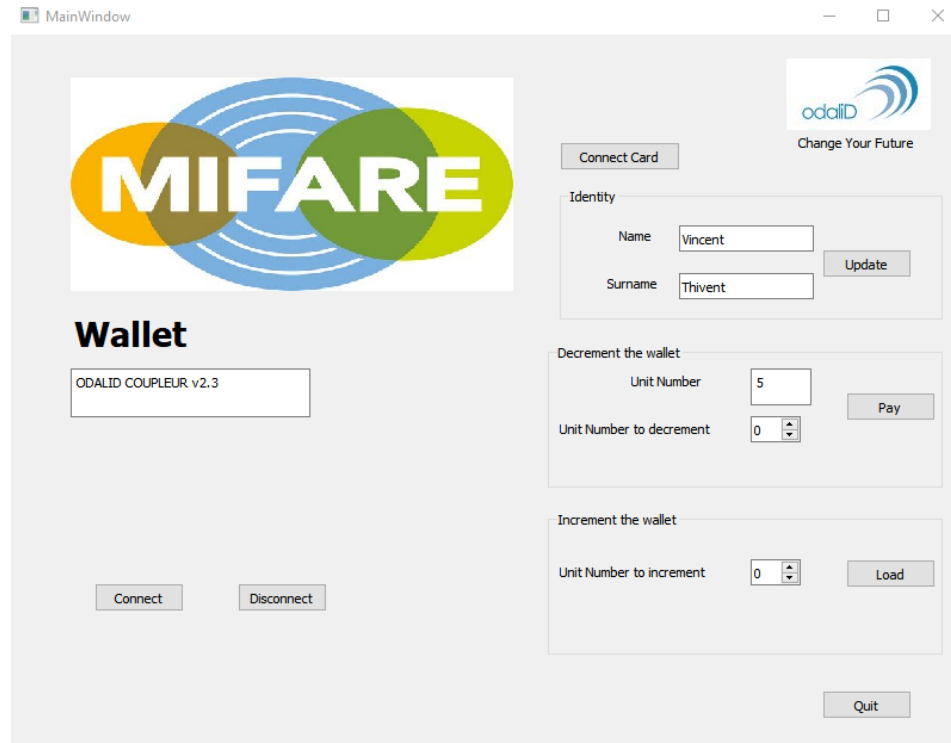


Figure 2 : Lecture de la carte sans-contact

La première zone permet de visualiser le nom et le prénom de la personne titulaire de la carte et de modifier ces informations directement à l'aide du bouton "Update".

La seconde zone nous permet de voir la valeur du compteur de la carte ainsi que de le décrémenter à l'aide du bouton "Pay".

Enfin, la dernière zone est prévue afin de pouvoir incrémenter la valeur du compteur en utilisant le bouton "Load".

Finalement, le bouton "Quit" permet de fermer l'application tout en éteignant les LED du lecteur et en déconnectant celui-ci.

En cas d'erreur de lecture ou d'écriture, un message d'erreur de couleur rouge s'affiche en dessous des boutons "Connect" et "Disconnect" afin d'informer l'utilisateur de l'erreur rencontrée. De plus, des bruits sonores émis par le lecteur s'activent lors des différentes opérations réalisées (connexion, lecture de carte, mise à jour de l'identité, etc) afin d'avertir l'utilisateur.

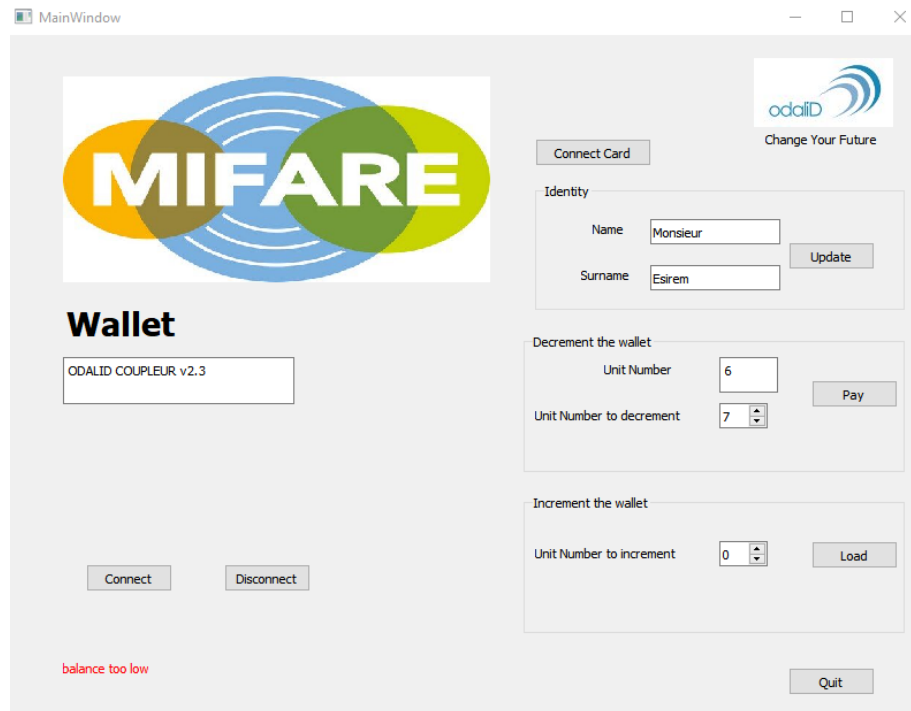


Figure 3 : Affichage de l'erreur à l'utilisateur

Conclusion

Ce projet de Communication Sans Fil sur la carte sans-contact MIFARE Classic nous a permis de mieux comprendre le fonctionnement de cette carte sans-contact. En développant notre propre programme pour interagir avec la carte, nous avons acquis une connaissance pratique des protocoles de communication sans fil et des mécanismes de sécurité utilisés dans ces systèmes. De plus, la création d'une interface utilisateur nous a donné l'occasion de réfléchir à la manière dont les informations peuvent être présentées de manière claire et intuitive. En somme, ce projet nous a offert une expérience précieuse dans l'application des concepts théoriques à un problème concret, tout en nous confrontant aux défis et aux complexités inhérents au développement de systèmes tels que celui-ci.