

Rapport de TP :

Communication sans fil

Carte sans contact MIFARE Classic

Réalisé par :

WALY Mouad

EL YOUSFI Mohammed

Encadré par :

M. THIVENT Vincent

Année scolaire :

2022/2023

1. Introduction :

Dans ce TP, nous allons développer une interface graphique pour la gestion de la carte sans-contact MIFARE Classic.

L'interface graphique doit pouvoir :

- Enrôler une carte MIFARE Classic d'origine.
- Incrémenter et décrémenter des unités dans la carte (avec la gestion d'un backup).
- Écrire et lire l'identité de la personne dans la carte (nom et prénom).
- Une information visuelle et/ou sonore du coupleur doit être faite à chaque transaction.

2. Architecture de la carte :

2.1 Identification :

Afin que les cartes soient lues par toutes les applications. Nous définirons une architecture commune à chaque carte. 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	Prénom
	8	« Identite »

L'authentification se fera avec les clés suivantes :

Key A : A0 A1 A2 A3 A4 A5

Key B : B0 B1 B2 B3 B4 B5

2.2 Compteur :

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

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

L'authentification se fera avec les clés suivantes :

Key A : C0 C1 C2 C3 C4 C5

Key B : D0 D1 D2 D3 D4 D5

3. Fonctionnalités techniques et logicielles :

Bouton connecter : permettra de se connecter à la carte.

```
void MaFenetre::on_connect_btn_clicked() {
    if (connected) {
        // cannot connect when already connected
        // disconnect first to refresh the connection
        on_disconnect_btn_clicked();
    }

    int16_t status = MI_OK;
    MonLecteur.Type = ReaderCDC;
    MonLecteur.device = 0;

    status = OpenCOM(&MonLecteur);
    if (status == 0) {
        connected = true;
        qDebug() << "Connected";
        status = Version(&MonLecteur);
        if (status == MI_OK) {
            ui->label_device->setText(MonLecteur.version);
        }
        status = RF_Power_Control(&MonLecteur, true, 0);
        status = LEDBuzzer(&MonLecteur, LED_GREEN_ON);
    }
}
```

Bouton qui permettra de détecter la carte :

```
void MaFenetre::on_card_btn_clicked()
{
    uint16_t status = MI_OK;
    uint8_t atq[2];
    uint8_t sak[1];
    uint8_t uid[12];
    uint16_t uidlen = 12;
    status = ISO14443_3_A_PollCard(&MonLecteur, atq, sak, uid, &uidlen);

    if (status == MI_OK) {
        LEDBuzzer(&MonLecteur, LED_GREEN_ON + LED_YELLOW_ON + BUZZER_ON);
    }
}
```

Bouton Mise à jour de l'identité :

Le prénom :

```
void MaFenetre::on_update_identity_btn_clicked()
{
    uint16_t status;

    uint8_t data[16];

    // first name
    QString firstname = ui->firstname_edit->text();
    int i = 0;
    while (i < firstname.size()) {
        data[i] = firstname.at(i).toLatin1();
        i++;
    }
    while (i < 16) {
        data[i] = 0;
        i++;
    }
    status = Mf_Classic_Write_Block(&MonLecteur, true, 9, data, key_b, 2);
    if(status != MI_OK) {
        qDebug() << "error: could not write first name";
    }
}
```

Le nom :

```
// last name
QString lastname = ui->lastname_edit->text();
i = 0;
while (i < lastname.size()) {
    data[i] = lastname.at(i).toLatin1();
    i++;
}
while (i < 16) {
    data[i] = 0;
    i++;
}
status = Mf_Classic_Write_Block(&MonLecteur, true, 10, data, key_b, 2);
if(status != MI_OK) {
    qDebug() << "error: could not write last name";
}
```

Bouton incrémenter : il permettra d'incrémenter le nombre d'unités à décrémente de la carte, il incrémente avec la valeur saisie dans le champ `raise_amount`.

```
void MaFenetre::on_raise_btn_clicked()
{
    uint32_t value = ui->raise_amount->value();

    uint16_t status;
    status = Mf_Classic_Increment_Value(&MonLecteur, true, 14, value, 14, key_b, 3);
    if (status == MI_OK) {
        qDebug() << "incremented correctly";

        // read the card to refresh the UI
        on_card_btn_clicked();
    } else {
        qDebug() << "could not write";
    }
}
```

Bouton décrémenter : il permettra de décrémenter le nombre d'unités avec la valeur value saisie dans le champ spent_amount.

```
void MaFenetre::on_spend_btn_clicked()
{
    uint32_t value = ui->spent_amount->value();

    uint16_t status;
    status = Mf_Classic_Decrement_Value(&MonLecteur, true, 14, value, 14, key_b, 3);
    if (status == MI_OK) {
        qDebug() << "decremented correctly";

        // read the card to update the UI
        on_card_btn_clicked();
    } else {
        qDebug() << "error: could not decrement";
    }
}
```

Bouton Quitter : permet de quitter l'application

```
void MaFenetre::on_leave_clicked()
{
    int16_t status = MI_OK;
    RF_Power_Control(&MonLecteur, FALSE, 0);
    status = LEDBuzzer(&MonLecteur, LED_OFF);
    status = CloseCOM(&MonLecteur);
    qApp->quit();
}
```

4) Conclusion :

Enfin, Nous obtenons la forme de l'interface graphique qui permet de lire l'identité, mettre à jour l'identité, incrémenter et décrémenter des valeurs de bloc de la carte sans contact.

Ce TP nous a donné une idée globale sur le fonctionnement des systèmes de télécommunications sans-fils.