COMPTE RENDU TP DE SYSTEMES DE TELECOMMUNICATIONS SANS-FILS ET APPLICATIONS

CARTE SANS CONTACT MIFARE-CLASSIC

Rédigé par :

KAMENI Gabriel

MASSUDOM Josepha

Sous l'encadrement de :

M. Vincent THIVENT

Année académique 2022/2023

PLAN

LISTE DES FIGURES

INTRODUCTION

Dans ce TP, nous allons étudier la carte MIFARE-CLASSIC. Il s'agit d'une carte qui répond à la norme ISO/IEC 14443 Type A, qui a une fréquence porteuse de 13,56Mhz et un débit de communication de 106Kbit/S. Nous allons principalement écrire les fonctions qui nous permettrons de lire et d'écrire sur la carte. Mais aussi de gérer le compteur.

I. Matériel et logiciel utilisés

Pour ce TP, nous avons utilisé le logiciel QtCreator afin de créer l'interface utilisateur nous permettant d'agir sur la carte à l'aide de fonctions écrites en C++. Nous avons également utilisé la carte Mifare-Classic elle-même que nous avons lu avec u lecteur de carte.

II. Développement de l'interface graphique

1. Fonctionnalités de l'interface graphique

Notre interface graphique doit nous permettre de répondre au diagramme de cas d'utilisation ci-dessous. Entre autres, l'écriture et la lecture de l'identité, l'incrémentation et la décrémentation du compteur, l'allumage de la led et du buzzer.

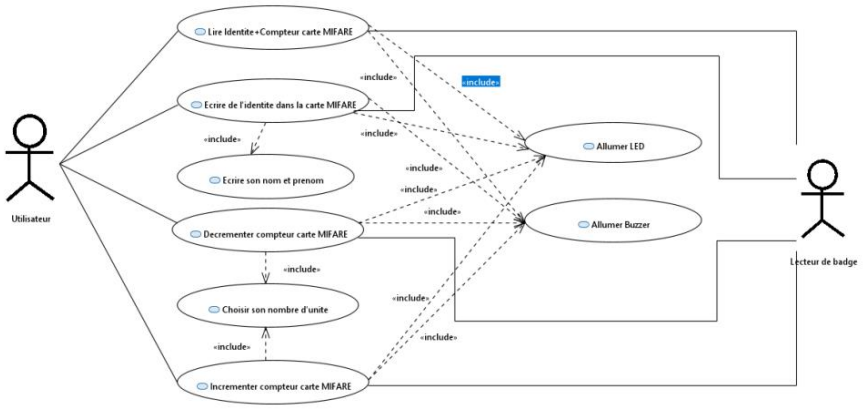


Figure 1: Fonctionnalités de l'interface graphique

2. Connexion

Le bouton Connect nous permet de nous connecter au lecteur de carte. Une fois la carte lue, nous pouvons dès à présent lui passer des données ou lire les données qui y sont écrites nous pouvons aussi nous déconnecter. Ci-dessous, les codes des boutons de connexion et déconnexion.

En effet, l'image ci-contre nous montre le code du bouton connect. On commence d'abord par initialiser le Reader ensuite on précise le port série, puis on envoie un status d'openCom et enfin on utilise la méthode RF_Power_Control pour générer un champ électromagnétique. Lorsque le Status est à Mi_Ok, cela stipule que la connexion est effective.

```
void MaFenetre::on_Connect_clicked() {
    int16_t status = MI_OK;
    MonLecteur.Type = ReaderCDC;
    MonLecteur.device = 0;
    status = OpenCOM(&MonLecteur);
    qDebug() << "OpenCOM" << status;

    status = Version(&MonLecteur);
    ui->Affichage->setText(MonLecteur.version);
    ui->Affichage->update();

   // QString Text = ui->fenetre_saisi->toPlainText();
    //qDebug() << "Text : " << Text;

    status = RF_Power_Control(&MonLecteur, TRUE, 0);

}
```

Figure 2: Connexion de la carte

3. Identification

Pour l'identification, nous avons utilisé deux blocs de données, les blocs 10 et 9.

Dans le bloc 10 nous écrivons le nom avec la clé B située dans le serrure Elément à index 2 et nous le lisons avec la clé B également située dans le serrure Elément à l'index 2.

Pour le prénom, nous l'écrivons dans le bloc 9 également avec la clé B et nous le lisons de même avec la clé A.

a. Lecture

Pour lire la carte, nous avons utilisé la fonction ISO14443_Pollcard à l'intérieur de laquelle nous initialisons les attributs. Puis, nous utilisons la fonction Mf_Classic_Read_Block dans laquelle nous passons en paramètre 9 pour le bloc du prénom et 2 pour le secteur.

Nous ajoutons par la suite dans la zone de texte réservée au prénom le contenu de notre variable que nous castons en char. Nous répétons cette opération pour le nom en changeant juste le secteur conformément à ce qui a été dit plus haut.

```
void MaFenetre::on_lire_clicked()
{
    uint8_t data[240] = {0};
    uint8_t atq[2];
    uint8_t sak[1];
    uint8_t uid[12];
    uint16_t uid_len = 12;

    int16_t status = MI_OK;

    status = ISO14443_3_A_PollCard(&MonLecteur, atq, sak, uid, &uid_len);
    status = Mf_Classic_Read_Block(&MonLecteur, TRUE, 9, data, AuthKeyA, 2);
    ui->nom->setText((char*)data);
    status = Mf_Classic_Read_Block(&MonLecteur, TRUE, 10, data, AuthKeyA, 2);
    ui->prenom->setText((char*)data);
    //qDebug() << *data;

   /* status = LEDBuzzer(&MonLecteur, LED_GREEN_ON+LED_YELLOW_ON+LED_RED_ON+LED_GREEN_ON);
    DELAYS_MS(1);
    status = LEDBuzzer(&MonLecteur, LED_GREEN_ON);*/

}
```

Figure 3: Lecture de la carte

b. Ecriture

Pour l'écriture, nous utilisons de nouveau Pollcard. On va donc récupérer le contenu de nos zones de texte à l'aide de la fonction toPlainText. On convertit d'abord l'information DataIn en utf8 afin de pouvoir l'envoyer à notre carte Mifare qui peut traiter ce type de données. En suite on utilise la fonction Mf_Classic_Write_Block afin d'écrire dans notre block.

```
void MaFenetre::on_mise_a_jour_clicked()
{
    uint8_t atq[2];
    uint8_t sak[1];
    uint8_t uid[12];
    uint16_t uid_len = 12;
    int16_t status = MI_OK;

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

    status = ISO14443_3_A_PollCard(&MonLecteur, atq, sak, uid, &uid_len);
    char DataIn1[16];
       strncpy(DataIn1, ui->nom->toPlainText().toUtf8().data(), 16);
    status = Mf_Classic_Write_Block(&MonLecteur, TRUE,  9,(uint8_t*)DataIn1, AuthKeyB, 2 );

    char DataIn[16];
       strncpy(DataIn, ui->prenom->toPlainText().toUtf8().data(), 16);
    status = Mf_Classic_Write_Block(&MonLecteur, TRUE,  10, (uint8_t*)DataIn, AuthKeyB, 2 );

}
```

Figure 4: Ecriture sur la carte

4. Compteur

Nous allons maintenant gérer l'incrémentation et la décrémentation du compteur.

a. Incrémentation

L'incrémentation nous permet d'augmenter notre compteur d'un certain nombre d'unités. Pour l'incrémentation, nous allons encore utiliser pollcard mais aussi Mf_Classic_Increment_Value à l'intérieure de laquelle nous passons le block 3 et le secteur 14 et la clé B puis nous utilisons Mf_Classic_Restore_Value afin de copier la valeur du tampon de transfert dans le block correspondant de la carte.

```
void MaFenetre::on_charger_clicked()
{
    uint32_t pValue = 0;
    uint8_t atq[2];
    uint8_t sak[1];
    uint8_t uid[12];
    uint16_t uid_len = 12;
    int16_t status = MI_OK;

    uint32_t ValeurEcrire = 0;
    ValeurEcrire = ui ->valeurIncrementer->value();

    status = ISO14443_3_A_PollCard(&MonLecteur, atq, sak, uid, &uid_len);
    //status = Mf_Classic_Write_Value(&MonLecteur, TRUE, 14, ValeurEcrire,  AuthKeyB, 3 );

    status = Mf_Classic_Increment_Value(&MonLecteur, TRUE, 14, ValeurEcrire, 13, AuthKeyB, 3);
    status =  Mf_Classic_Restore_Value(&MonLecteur, TRUE, 13, 14, AuthKeyB, 3);

    //AFFICAGE DE LA VALEUR FINAL D'UNITE
    status =  Mf_Classic_Read_Value(&MonLecteur, TRUE, 14, &pValue, AuthKeyA, 3 );
    ui->nombreUnite->setText(QString::number(pValue));


    status = LEDBuzzer(&MonLecteur, LED_GREEN_ON+LED_YELLOW_ON+LED_RED_ON+LED_GREEN_ON);
    DELAYS_MS(500);
    status = LEDBuzzer(&MonLecteur, LED_RED_ON);
```

Figure 5: Incrémentation du compteur

b. Décrémentation

Pour la décrémentation, on réalise les mêmes actions mais avec Mf_Classic_Decrement_Value.

```
void MaFenetre::on_Buy_clicked()
{
    uint32_t pValue = 0;
    uint8_t atq[2];
    uint8_t sak[1];
    uint8_t uid[12];
    uint16_t uid_len = 12;
    int16_t status = MI_OK;

    uint32_t Valeurdecrementer = 0;
    Valeurdecrementer = ui ->valeurDecrementer->value();
    status = ISO14443_3_A_PollCard(&MonLecteur, atq, sak, uid, &uid_len);

    status =  Mf_Classic_Read_Value(&MonLecteur, TRUE, 14, &pValue, AuthKeyA, 3 );


    status =  Mf_Classic_Decrement_Value(&MonLecteur, TRUE, 14, Valeurdecrementer, 13, AuthKeyA, 3 );
    status =  Mf_Classic_Restore_Value(&MonLecteur, TRUE, 13, 14, AuthKeyA, 3);
     ui->nombreUnite->setText(QString::number(pValue));

     status = LEDBuzzer(&MonLecteur, LED_GREEN_ON+LED_YELLOW_ON+LED_RED_ON+LED_GREEN_ON);
     DELAYS_MS(500);
     status = LEDBuzzer(&MonLecteur, LED_RED_ON);

}
```

Figure 6: Décrémentation du compteur

5. Led et Buzzer

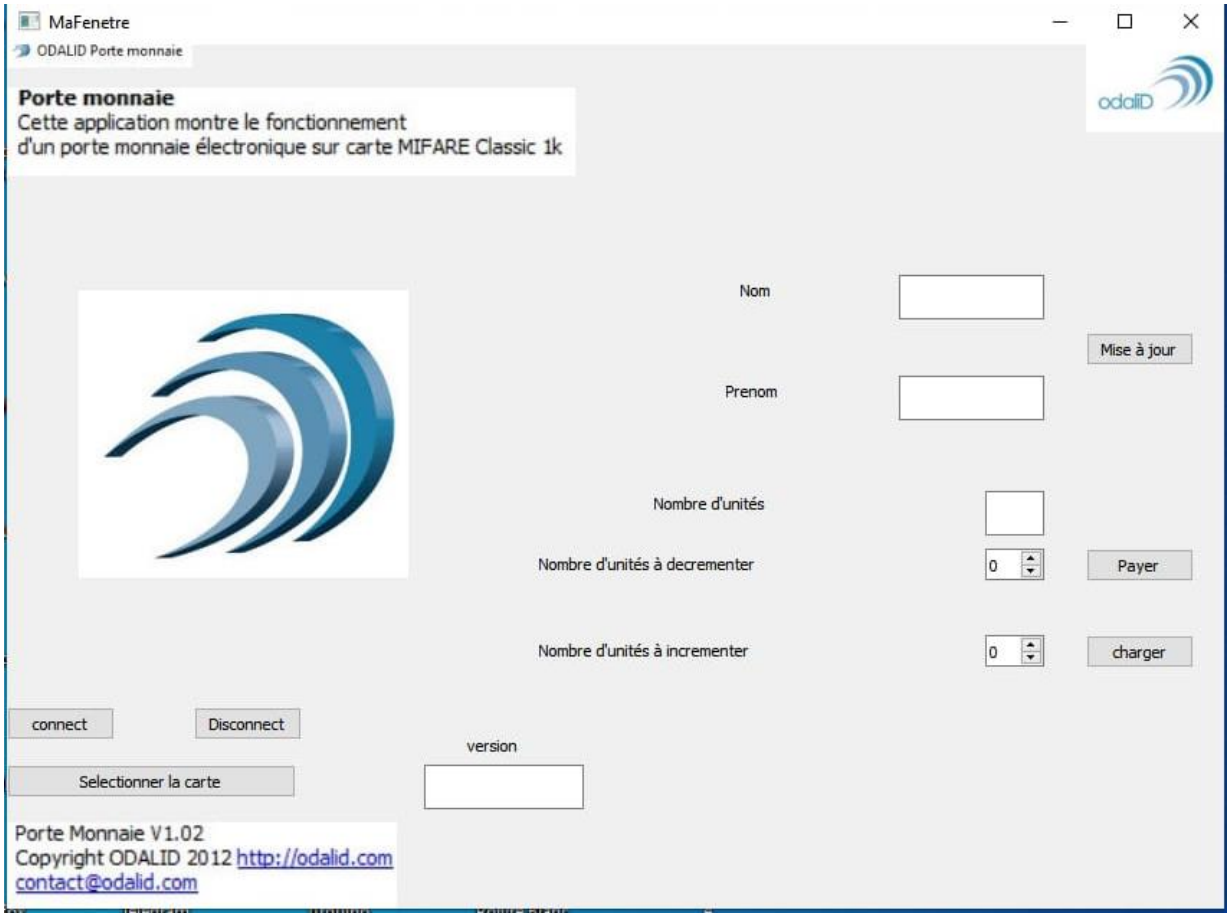
Ici, nous configurons l'allumage des led et buzzer pour chacune des opérations effectuées sur la carte conformément au diagramme de cas d'utilisation demandé.

```
status = LEDBuzzer(&MonLecteur, LED_GREEN_ON+LED_YELLOW_ON+LED_RED_ON+LED_GREEN_ON);
DELAYS_MS(500);
status = LEDBuzzer(&MonLecteur, LED_RED_ON);
```

Figure 7: Led et buzzer

III. Interface graphique

Nous avons donc notre interface graphique ci-après :



CONCLUSION

En somme, nous avons pu au cours de ces séances de TP comprendre le fonctionnement des cartes sans fil et des lecteurs de ces cartes. Nous avons également pu utiliser le logiciel QtCreator qui est un bon outil de création d'interface graphique.