

<RFID 출입관리 시스템 설명서>

Index:

Name:

UID:

<Picture>

DB STATUS: DB Connection Success

Music Play

DB 조회 1회 읽기 DB값 수정 Start Reading

버튼은 5개가 있습니다. dlg는 총 3개를 사용하였고, RFID_REGISTER, SELRes, RFIDDlg 3개의 클래스를 사용하였습니다.

*DB Select를 해주는 **DB 조회 버튼**, 새로운 Dlg창이 뜨며 DB select 결과를 새로운 Dlg창의 edit control에 출력해준다. 왼쪽은 출력결과 오른쪽은 실제 db에 들어있는 데이터

SELECT RESULT

<SELECT RESULT>

1 7b2168cb500104e0 lena lena.jpg
2 8a4c69cb500104e0 girl girl.jpg
3 31bcbfaf cap cap.jpg
4 818ece5b beard beard.jpg

확인 취소

출입 관리 시스템

Index:

Name:

UID:

<Picture>

DB STATUS: DB Select SUCCESS

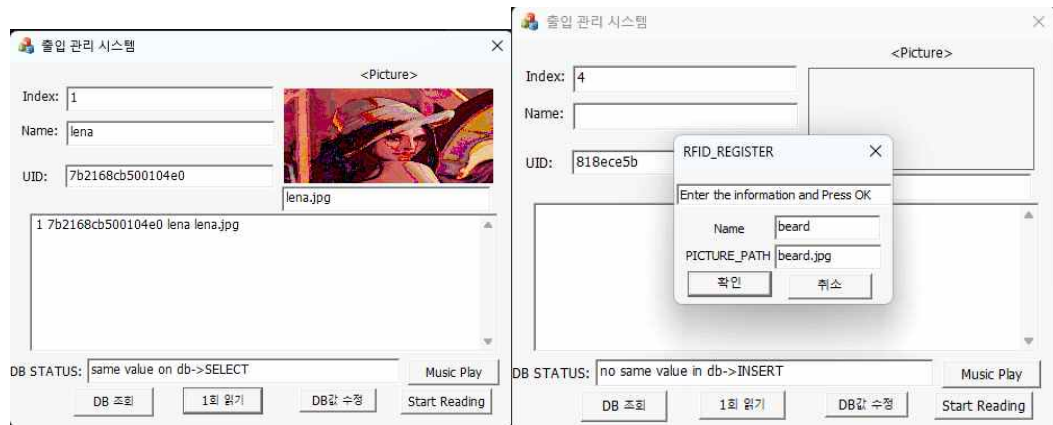
Music Play

DB 조회 1회 읽기 DB값 수정 Start Reading

num	uid	name	image
1	7b2168cb500104e0	lena	lena.jpg
2	8a4c69cb500104e0	girl	girl.jpg
3	31bcbfaf	cap	cap.jpg
4	818ece5b	beard	beard.jpg

*RFID를 1번 읽어주는 **1회 읽기 버튼**, 해당 UID에 해당되는 내용을 DB에서 select해오고 parsing 후 각각 edit control에 출력해줍니다. db에는 사진의 경로값을 저장하고 사진은 해

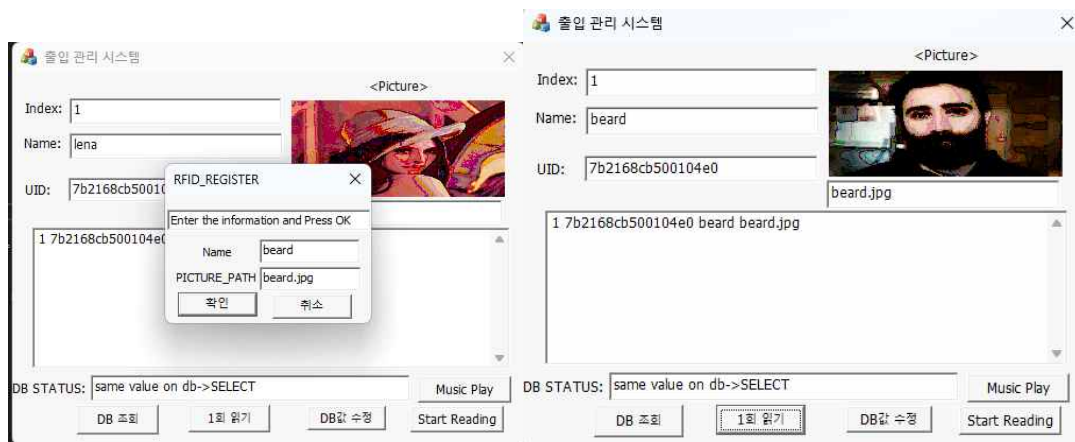
당 경로에 접근해서 출력해줍니다. 만약 해당 uid가 이미 db에 등록되어 있다면 select로 값을 읽어오고, 해당 uid가 db에 등록되어 있지 않다면, 새로운 dlg창을 띄워 name과 image 경로를 입력받고 insert into문을 통해 db에 값을 넣어주게 된다.



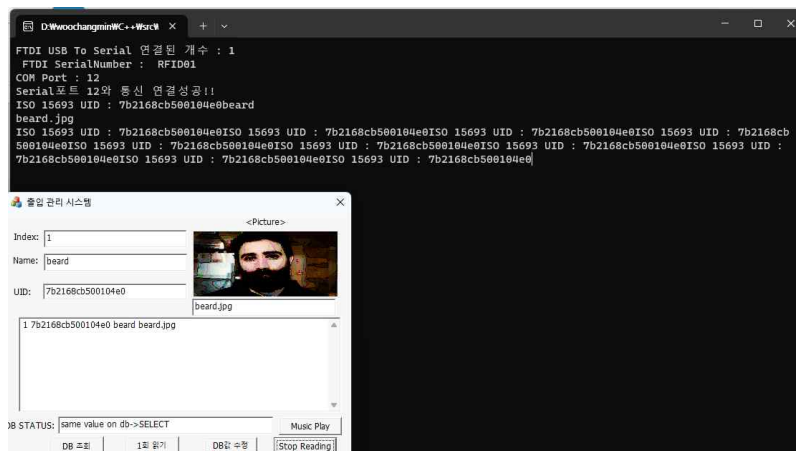
입력 뒤 1회읽기로 읽어들

이면 다음과 같이 잘 나오는 것을 확인 할 수 있다.

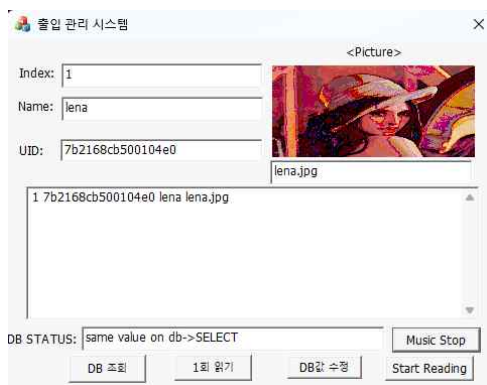
*1회 읽은 후 DB에 등록 되어 있는 DB값을 update 해주는 **DB값 수정 버튼**, 버튼을 누르면 새로운Dlg창이 뜨고 수정할 값을 입력한 후 다시 읽어주면 값이 바뀌어 있는 것을 확인 할 수 있음.



*쓰레드를 활용해서 연속적으로 읽어주는 **Start Reading** 버튼, 한 번 누르면 연속적으로 RFID를 읽어주고 버튼명은 Stop Reading으로 바뀝니다. Stop Reading 버튼을 누르면 연속 읽기를 중지하고 버튼명이 Start Reading이 됩니다.



*wav파일을 실행시켜주는 **Music Play** 버튼, 한번 누르면 음악이 재생되고, 버튼명이 Music Stop으로 바뀝니다. Music Stop 버튼을 누르면 음악이 멈추고 다시 Music Play 버튼이 됩니다.



<소스코드>
(RFIDDlg.cpp)

```
// RFIDDlg.cpp: 구현 파일
//
//1. 입력 새로운 dlg로 받아서 새 정보 등록하는거 하나
//2. 입력 새로운 dlg로 받아서 원래 정보 수정하는거 하나
//3. 쓰레드 활용 연속읽기(수정 필요)
//4. 여러개 읽기
```

```
#include "is_d2xx.h"
#include "pch.h"
#include "framework.h"
#include "RFID.h"
#include "RFIDDlg.h"
#include "SELRes.h"
#include "afxdialogex.h"
```

```
#include "mmsystem.h"
#pragma comment(lib,"winmm.lib")
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#endif
```

```
UINT ThreadForCounting(LPVOID param);
```

```
MYSQL* conn;
MYSQL* conn_result;
unsigned int timeout_sec = 1;
//char query[1024];
MYSQL_RES* result;
MYSQL_ROW row;
char select_num_query1[30] = { 0 };
int num = 0;
CString insert_query = _T("");
//char insert_query1[1024] = { 0 };
CString temp2 = _T("");
```

```
CString name;  
CString db_select_result = _T("");  
MYSQL_ROW m_mysqlRow;
```

```
CString m_str_Name;  
CString m_str_pic_path;
```

```
CString m_UID=_T("");
```

```
// 응용 프로그램 정보에 사용되는 CAboutDlg 대화 상자입니다.
```

```
class CAboutDlg : public CDialogEx  
{  
public:
```

```
    CAboutDlg();
```

```
// 대화 상자 데이터입니다.
```

```
#ifdef AFX_DESIGN_TIME
```

```
    enum { IDD = IDD_ABOUTBOX };
```

```
#endif
```

```
protected:
```

```
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 지원입  
니다.
```

```
// 구현입니다.
```

```
protected:
```

```
    DECLARE_MESSAGE_MAP()
```

```
};
```

```
CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
```

```
{  
}
```

```
void CAboutDlg::DoDataExchange(CDataExchange* pDX)
```

```
{  
    CDialogEx::DoDataExchange(pDX);  
}
```

```
BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
```

```
END_MESSAGE_MAP()
```

```
// CRFIDDlg 대화 상자
```

```
CRFIDDlg::CRFIDDlg(CWnd* pParent /*=nullptr*/)  
    : CDialogEx(IDD_RFID_DIALOG, pParent)
```

```
{  
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);  
    m_isWorkingThread = true;  
    m_pThread = NULL;  
    m_isWorkingThread = false;  
    //m_pThread = AfxBeginThread(ThreadForCounting, this);  
}
```

```
CRFIDDlg::~CRFIDDlg()
```

```
{  
    //연결 해제 추가  
  
    m_isWorkingThread = false;  
  
    WaitForSingleObject(m_pThread->m_hThread, 5000);  
  
    is_RfOff(ftHandle); // 무선파워를 끕어요.  
  
    //USB 포트를 Close  
    if (is_Close(ftHandle) == IS_OK)  
    {  
        printf("연결을 닫습니다. ");  
    }  
  
    m_isWorkingThread = false;  
  
    WaitForSingleObject(m_pThread->m_hThread, 5000);  
}
```

```
void CRFIDDlg::DoDataExchange(CDataExchange* pDX)  
{
```

```

CDialogEx::DoDataExchange(pDX);
// DDX_Text(pDX, IDC_EDIT1, m_strRfid);
DDX_Control(pDX, IDC_PIC, m_picture_control);
// DDX_Text(pDX, IDC_EDIT4, m_str_DBstatus);
DDX_Control(pDX, IDC_EDIT4, m_ctrl_DBStatus);
DDX_Control(pDX, IDC_EDIT1, m_ctrl_UID);
DDX_Control(pDX, IDC_EDIT5, m_ctrl_select_result);
DDX_Control(pDX, IDC_EDIT3, m_ctrl_Index);
DDX_Control(pDX, IDC_EDIT6, m_ctrl_pic_name);
DDX_Control(pDX, IDC_EDIT2, m_ctrl_Name);
}

BEGIN_MESSAGE_MAP(CRFIDDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGON()
    ON_BN_CLICKED(IDC_BUTTON1, &CRFIDDlg::OnDBSelect)
    ON_BN_CLICKED(IDC_BUTTON2, &CRFIDDlg::OnReadOnce)
    ON_BN_CLICKED(IDC_BUTTON3, &CRFIDDlg::OnDBUpdate)
    ON_BN_CLICKED(IDC_BUTTON4, &CRFIDDlg::OnMusicPlay)
    ON_BN_CLICKED(IDC_BUTTON5, &CRFIDDlg::OnReadManyTime)

END_MESSAGE_MAP()

// CRFIDDlg 메시지 처리기

BOOL CRFIDDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 시스템 메뉴에 "정보..." 메뉴 항목을 추가합니다.

    // IDM_ABOUTBOX는 시스템 명령 범위에 있어야 합니다.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;

```

```

        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING,          IDM_ABOUTBOX,
strAboutMenu);
        }
    }

    // 이 대화 상자의 아이콘을 설정합니다. 응용 프로그램의 주 창이 대화 상자가 아
닐 경우에는
    // 프레임워크가 이 작업을 자동으로 수행합니다.
    SetIcon(m_hIcon, TRUE);           // 큰 아이콘을 설정합니다.
    SetIcon(m_hIcon, FALSE);        // 작은 아이콘을 설정합니다.

    // TODO: 여기에 추가 초기화 작업을 추가합니다.
    if (is_GetDeviceNumber(&usbnumber) == IS_OK)
    {
        printf("FTDI USB To Serial 연결된 개수 : %d\n", usbnumber);
        if (is_GetSerialNumber(0, readSerialNumber) == IS_OK)
        {
            if (memcmp(changeSerialNumber, readSerialNumber,
sizeof(changeSerialNumber) != 0))
            {
                if (is_SetSerialNumber(0, changeSerialNumber) ==
IS_OK)
                {
                    printf(" USB To Serial Number 를 변경 하였
습니다.\n");
                    printf(" FTDI SerialNumber : %s \n",
changeSerialNumber);
                }
            }
            else
                printf(" FTDI SerialNumber : %s \n",
changeSerialNumber);
        }
    }
}

```



```

//열린 포트번호로 연결
unsigned long portNumber;
if (is_GetCOMPort_NoConnect(0, &portNumber) == IS_OK)
{
    printf("COM Port : %d\n", portNumber);
}
if (is_OpenSerialNumber(&ftHandle, readSerialNumber, 115200) != IS_OK)
{
    printf("USB To Serial과 통신 연결 실패\n");
    //return -1;
}
else {
    printf("Serial포트 %d와 통신 연결성공!! \n", portNumber);
}
Sleep(100);
flag_r = 0;
conn = mysql_init(NULL);
mysql_options(conn, MYSQL_OPT_CONNECT_TIMEOUT, &timeout_sec);
conn_result = mysql_real_connect(conn, "127.0.0.1", "root", "dnckdals1!",
"mysql", 3306, NULL, 0);
//비밀번호 바꿔주세요!!

if (NULL == conn_result)
{
    m_ctrl_DBStatus.SetWindowTextA(_T("DB Connection Fail"));
}
else
{
    m_ctrl_DBStatus.SetWindowTextA(_T("DB Connection Success"));
}

return TRUE; // 포커스를 컨트롤에 설정하지 않으면 TRUE를 반환합니다.
}

void CRFIDDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFFF0) == IDM_ABOUTBOX)

```

```

    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

```

// 대화 상자에 최소화 단추를 추가할 경우 아이콘을 그리려면
// 아래 코드가 필요합니다. 문서/뷰 모델을 사용하는 MFC 애플리케이션의 경우에는
// 프레임워크에서 이 작업을 자동으로 수행합니다.

```

void CRFIDDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 그리기를 위한 디바이스 컨텍스트입니다.

        S e n d M e s s a g e ( W M _ I C O N E R A S E B K G N D ,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 클라이언트 사각형에서 아이콘을 가운데에 맞춥니다.
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // 아이콘을 그립니다.
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
}

```

// 사용자가 최소화된 창을 끄는 동안에 커서가 표시되도록 시스템에서

```
// 이 함수를 호출합니다.
HCURSOR CRFIDDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}
```

```
//쓰레드 활용
UINT ThreadForCounting(LPVOID param)
{
    CRFIDDlg* pMain = (CRFIDDlg*)param;
    CString temp, temp1 = _T("");

    while (pMain->m_isWorkingThread)
    {
        Sleep(1000);
        pMain->OnReadOnce();
    }

    return 0;
}
```

```
void CRFIDDlg::OnDBSelect()
{
    m_ctrl_Index.SetWindowTextA(_T(""));
    m_ctrl_Name.SetWindowTextA(_T(""));
    m_ctrl_UID.SetWindowTextA(_T(""));
    m_ctrl_pic_name.SetWindowTextA(_T(""));

    m_ctrl_DBStatus.SetWindowTextA(_T("DB Select"));
    char query[1024];
    sprintf_s(query, 1024, "SELECT * FROM rfid");//아까전에 만들어줬던 db 내 테이블 이름
```

```

// Send Query
if (mysql_query(conn, query))
{
    m_ctrl_DBStatus.SetWindowTextA(_T("DB Select FAIL"));
    //cout << "SELECT Fail" << endl;
}
else
{
    db_select_result = _T("");
    CString For_input_Test=_T("");
    m_ctrl_DBStatus.SetWindowTextA(_T("DB Select SUCCESS"));
    // Get Response
    result = mysql_store_result(conn);

    int fields = mysql_num_fields(result);    // 필드 갯수 구함

    while (row = mysql_fetch_row(result))    // 모든 레코드 탐색
    {
        for (int i = 0; i < fields; i++)    // 각각의 레코드에서 모든 필드 값 출력
        {
            //printf("%s ", row[i]);
            db_select_result += row[i];
            db_select_result += " ";
            //For_input_Test
            //cout << row[i] << " ";
        }
        db_select_result += "\r\n";
        //printf("\n");
    }
    //m_ctrl_select_result.SetWindowTextA(db_select_result);
    SELRes sr;
    sr.DoModal();

}

}

void CRFIDDlg::OnReadOnce()
{

```

```

CString temp = _T("");
CString temp2 = _T(",");
CString temp3 = _T("");
int i;
char cnum[4] = { 0 };

m_UID = _T("");

//ISO15693모드로 읽기( 싱글모드 읽기 )
if (flag_r == 0)
{
    int read_flag = 0;
    //---db 데이터 갯수 가져오기

    sprintf_s(select_num_query1, 30, "SELECT MAX(num) FROM
rfid");

    if (mysql_query(conn, select_num_query1))
    {
        printf("SELECT FAIL\n");
    }
    // Get Response
    result = mysql_store_result(conn);
    row = mysql_fetch_row(result);
    sprintf(cnum, "%s", row[0]);
    num = atoi(cnum);
    CString number;
    number.Format(_T("%d"), num);
    //
    //num= index 최대값

    //-----db data 갯수 갖고오기 끝

    if (is_WriteReadCommand(ftHandle, CM1_COMMON,
CMD2_COMMON_ALL_UID_READ,
writeLength, wirteData, &readLength, readData) == IS_OK) {

        if (readLength == 8)//정상적으로 읽힌 단일 데이터라면
        {
            m_UID = _T("");

```

```

printf("ISO 15693 UID : ");

insert_query.Format(_T("INSERT INTO
rfid(num,uid,name,image) VALUES(%d,"), num + 1);
for (i = 0; i < readLength; i++)
{
    temp.Format(_T("%02x"), readData[i]);
    m_UID += temp;//temp1이 for문 끝나면 uid
전체를 갖고있게됨.

    printf("%02x", readData[i]);
}
read_flag = 1;
}
else if (readLength == 4)//정상적으로 읽힌 단일 데이터라면
{
    printf("ISO 14443AB UID: ");
    m_UID = _T("");
    insert_query.Format(_T("INSERT INTO
rfid(num,uid,name,image) VALUES(%d,"), num+1);
    for (int i = 0; i < readLength; i++)
    {
        temp.Format(_T("%02x"), readData[i]);
        m_UID += temp;//temp1이 for문 끝나면 uid
전체를 갖고있게됨.

        printf("%02x", readData[i]);
    }
    read_flag = 1;
}
if(read_flag)//정상적으로 14443이나 15693 데이터가 읽혔다면
{
    read_flag = 0;
    m_ctrl_UID.SetWindowTextA(m_UID);
    //=====db에 해당 uid가 있는지 체크
    CString find_uid_query = _T("SELECT * FROM rfid
WHERE uid='");

    find_uid_query += m_UID;
    find_uid_query += "'";
    const char* tmp;
    tmp = (CStringA)find_uid_query;
    if (mysql_query(conn, tmp))
    {

```

```

        printf("SQL Fail");
    }
    else
    {
        // Get Response
        result = mysql_store_result(conn);
        if (!mysql_fetch_row(result)) { //동일한 값이 없
다면 insert

m_ctrl_DBStatus.SetWindowTextA(_T("no same value in db->INSERT"));

insert_query.Insert(insert_query.GetLength(), m_UID); //uid 전체 넘겨주기 배열 0부터니
까 위에 쿼리 길이-1

        //insert_query += temp1;
        number.Format(_T("%d"), num+1);

m_ctrl_Index.SetWindowTextA(number);

        RFID_REGISTER RR;
        RR.DoModal();

insert_query.Insert(insert_query.GetLength(), temp2);

        temp3 = m_str_Name;
        temp3 += ", ";
        temp3 += m_str_pic_path;
        temp3 += " ";

insert_query.Insert(insert_query.GetLength(), temp3);
        cout << insert_query << endl;

        printf("\n");

        const char* insert_query1;
        insert_query1 =

(CStringA)insert_query;

        //for (i = 0; i <
insert_query.GetLength(); i++)

        //{

```

```

insert_query[i];

insert_query1[i]);//(char*)(LPCTSTR)

//      insert_query1[i]          =

//      p r i n t f ( " % c " ,

//}

printf("\n");
printf("\n");

// Send Query
if          (mysql_query(conn,

insert_query1))//insert_query1

{
    printf("SQL Fail");
}
// Get Response
result = mysql_store_result(conn);

}
else //동일한 값이 있다면 select
{

m_ctrl_DBStatus.SetWindowTextA(_T("same value on db->SELECT"));

//m_ctrl_UID.SetWindowTextA(temp1);

CString      select_num_query      =

_T("SELECT * FROM rfid WHERE uid=");

select_num_query += m_UID;
select_num_query += "";
const char* tmp;

tmp = (CStringA)select_num_query;
//printf("%s", tmp);
// Send Query
if          (mysql_query(conn,

tmp))//select_uid_query

{
    printf("SQL Fail");
}

```



```

    }
    // Get Response
    else {
        result =

mysql_store_result(conn);

    }

    int m_llResCount =

mysql_num_fields(result);

    if (m_llResCount == 0)
    {
    }
    else
    {
        CString s_row_data = _T("");
        while ((m_mysqlRow =

mysql_fetch_row(result)))

        {
            for (size_t i = 0; i <

m_llResCount; i++)

            {
                C S t r i n g

                s_row_data

                s_row_data

            }

m_ctrl_select_result.SetWindowTextA(s_row_data);

C S t r i n g

s_index(m_mysqlRow[0]);

C S t r i n g

s_name(m_mysqlRow[2]); //name

C S t r i n g

s_image_url(m_mysqlRow[3]); //image url

//std::cout << "\n"

<< strTest2 << std::endl;

//std::cout << "\n"

<< s_image_url << std::endl;

```

```
m_ctrl_Index.SetWindowTextA(s_index);
```

```
m_ctrl_Name.SetWindowTextA(s_name);
```

```
m_ctrl_pic_name.SetWindowTextA(s_image_url);
```

```
                                CRect rect;//픽처 컨트롤의 크기를 저장할 CRect 객체
```

```
m_picture_control.GetWindowRect(rect);//GetWindowRect를 사용해서 픽처 컨트롤의 크기를 받는다.
```

```
                                CDC* dc; //픽처 컨트롤의 DC를 가져올 CDC 포인터  
                                dc =
```

```
m_picture_control.GetDC(); //픽처 컨트롤의 DC를 얻는다.
```

```
                                CImage image;//불러  
오고 싶은 이미지를 로드할 CImage
```

```
image.Load(s_image_url);//이미지 로드
```

```
image.StretchBlt(dc->m_hDC, 0, 0, rect.Width(), rect.Height(), SRCCOPY);//이미지를  
픽처 컨트롤 크기로 조정
```

```
                                ReleaseDC(dc);//DC  
해제
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```

void CRFIDDlg::OnDBUpdate()
{
    //UID는 직접 센싱
    RFID_REGISTER RR;
    RR.DoModal();

    cout << m_str_Name << endl;
    cout << m_str_pic_path << endl;

    CString s_m_Name = _T(""), s_m_UID = _T(""), s_m_pic_Name =
_T(""); //modify용 변수들
    m_ctrl_UID.GetWindowTextA(s_m_Name);
    CString modify_query = _T("");
    //UPDATE rfid SET name='wcm', image='wcow.jpg' WHERE
uid='8a4c69cb500104e0'
    modify_query.Format(_T("UPDATE rfid SET name="));
    modify_query += m_str_Name;
    modify_query += ", image=";
    modify_query += m_str_pic_path;
    modify_query += " WHERE uid=";
    modify_query += m_UID;
    modify_query += "";

    const char* tmp;
    tmp = (CStringA)modify_query;

    // Send Query
    if (mysql_query(conn, tmp)) //select_uid_query
    {
        printf("SQL Fail");
    }
    // Get Response
    else {
        result = mysql_store_result(conn);
    }
}

```

```

}

void CRFIDDlg::OnMusicPlay()
{
    CString caption;
    GetDlgItemText(IDC_BUTTON4, caption);
    CString szSoundPath = _T("./bgm.wav");
    if (caption == _T("Music Play"))
    {
        SetDlgItemText(IDC_BUTTON4, _T("Music Stop"));
        PlaySound(szSoundPath, AfxGetInstanceHandle(), SND_ASYNC |
SND_LOOP); // 무한
    }
    else
    {
        SetDlgItemText(IDC_BUTTON4, _T("Music Play"));
        PlaySound(NULL, AfxGetInstanceHandle(), NULL); // 정지
    }

    /*
    {
        //CM1_ISO15693
        // CM2_ISO15693_MULTIPLE_BLOCK_READ
        //CMD2_COMMON_ALL_UID_READ
        //CM2_ISO15693_WRITE_AFI

        CString temp1 = _T("");
        char temp2[100] = { 0 };
        int n = 0;
        if (is_WriteCommand(ftHandle, CM1_COMMON,
CMD2_COMMON_ALL_UID_READ, writeLength, wirteData) == IS_OK)
        {
            //스레드작업으로 처리할때 편리하게사용할수있습니다.

            if (is_ReadExCommand(ftHandle, &cmd1, &cmd2, &readLength,
readData) == IS_OK)
            {
                int i;

```

```

        printf("Command1 = %x, Command2 = %x\n", cmd1, cmd2);
        printf("UID : ");
        for (i = 0; i < readLength; i++)
        {

            printf("%2x ", readData[i]);
            temp1 += readData[i];

        }
        temp1 += '\n';
    }
    printf("\n%s", temp2);

}
} //여러번 읽기
*/
}

```

```

void CRFIDDlg::OnReadManyTime()
{
    //쓰레드
    CString caption;
    GetDlgItemText(IDC_BUTTON5, caption);

    if (caption == _T("Start Reading"))
    {
        m_isWorkingThread = true;

        m_pThread = AfxBeginThread(ThreadForCounting, this);

        SetDlgItemText(IDC_BUTTON5, _T("Stop Reading"));
    }
    else
    {

        m_isWorkingThread = false;
    }
}

```

```

        WaitForSingleObject(m_pThread->m_hThread, 100);

        SetDlgItemText(IDC_BUTTON5, _T("Start Reading"));
    }

}

```

(RFIDDlg.h)

```

// RFIDDlg.h: 헤더 파일
//

```

```

#pragma once
#include "is_d2xx.h"
#include <afxwin.h>

```

```

// CRFIDDlg 대화 상자
class CRFIDDlg : public CDialogEx
{
// 생성입니다.
public:
    CRFIDDlg(CWnd* pParent = nullptr);    // 표준 생성자입니다.
    ~CRFIDDlg();

```

```

    IS_HANDLE ftHandle = 0;
    char readSerialNumber[100] = "COM07";
    char changeSerialNumber[100] = "RFID01";
    short usbnumber;
    unsigned char wirteData[1024];
    unsigned short writeLength = 2;
    unsigned char readData[1024];
    unsigned short readLength = 2;
    unsigned char cmd1=0;

```

```

        unsigned char cmd2=0;
        BOOL flag_r=0;

        CWinThread* m_pThread;
        bool m_isWorkingThread;

// 대화 상자 데이터입니다.
#ifdef AFX_DESIGN_TIME
        enum { IDD = IDD_RFID_DIALOG };
#endif

protected:
        virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 지원입
        니다.

// 구현입니다.
protected:
        HICON m_hIcon;

        // 생성된 메시지 맵 함수
        virtual BOOL OnInitDialog();
        afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
        afx_msg void OnPaint();
        afx_msg HCURSOR OnQueryDragIcon();
        DECLARE_MESSAGE_MAP()
public:
        afx_msg void OnDBSelect();
        afx_msg void OnReadOnce();
        afx_msg void OnDBUpdate();
        afx_msg void OnMusicPlay();
        afx_msg void OnReadManyTime();

        CStatic m_picture_control;
        CEdit m_ctrl_DBStatus;
        CEdit m_ctrl_UID;
        CEdit m_ctrl_select_result;
        CEdit m_ctrl_Index;
        CEdit m_ctrl_pic_name;
        CEdit m_ctrl_Name;

```

```

/*
void Set_str_Name(CString str_Name)
{

    //m_str_Name = str_Name;
    //char* temp;
    //strcpy(temp, (char*)(LPCTSTR)str_Name);
    //다시 CHAR -> CSTRING

}
void Set_str_Pic_name(CString Pic_name)
{

    //m_str_pic_path = Pic_name;

}
*/
};

```

(RFID_REGISTER.cpp)

```

// RFID_REGISTER.cpp: 구현 파일
//

#include "pch.h"
#include "RFID.h"
#include "afxdialogex.h"
#include "RFID_REGISTER.h"

extern CString m_str_Name;
extern CString m_str_pic_path;

// RFID_REGISTER 대화 상자

IMPLEMENT_DYNAMIC(RFID_REGISTER, CDialog)

RFID_REGISTER::RFID_REGISTER(CWnd* pParent /*!=nullptr*/)
    : CDialog(IDD_RFID_REGISTER, pParent)
{

}

RFID_REGISTER::~RFID_REGISTER()

```



```
{  
}
```

```
void RFID_REGISTER::DoDataExchange(CDataExchange* pDX)  
{  
    CDialog::DoDataExchange(pDX);  
    DDX_Control(pDX, IDC_EDIT1, m_ctrl_RName);  
    DDX_Control(pDX, IDC_EDIT2, m_ctrl_Rpic_path);  
    DDX_Control(pDX, IDC_EDIT3, m_ctrl_Guideline);  
}
```

```
BEGIN_MESSAGE_MAP(RFID_REGISTER, CDialog)  
    ON_BN_CLICKED(IDOK, &RFID_REGISTER::OnBnClickedOk)  
END_MESSAGE_MAP()
```

```
// RFID_REGISTER 메시지 처리기
```

```
void RFID_REGISTER::OnBnClickedOk()  
{  
    CString t_Name = _T("");  
    CString t_Pic_Path = _T("");  
    //  
  
    m_ctrl_RName.GetWindowTextA(m_str_Name);  
    m_ctrl_Rpic_path.GetWindowTextA(m_str_pic_path);  
  
    CDialog::OnOK();  
}
```

```
BOOL RFID_REGISTER::OnInitDialog()  
{  
    CDialog::OnInitDialog();  
  
    // TODO: 여기에 추가 초기화 작업을 추가합니다.  
    m_ctrl_Guideline.SetWindowTextA(_T("Enter the information and Press OK"));
```

```

    return TRUE; // return TRUE unless you set the focus to a control
    // 예외: OCX 속성 페이지는 FALSE를 반환해야 합니다.
}

```

(RFID_REGISTER.h)

```

#pragma once
#include "afxdialogex.h"
#include "RFIDDlg.h"

// RFID_REGISTER 대화 상자

class RFID_REGISTER : public CDialog
{
    DECLARE_DYNAMIC(RFID_REGISTER)

public:
    RFID_REGISTER(CWnd* pParent = nullptr); // 표준 생성자입니다.
    virtual ~RFID_REGISTER();

    // 대화 상자 데이터입니다.
#ifdef AFX_DESIGN_TIME
        enum { IDD = IDD_RFID_REGISTER };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV 지원입
    니다.

    DECLARE_MESSAGE_MAP()

public:
    //      CEdit m_Edit_Name;
    //      CEdit m_Edit_UID;
    afx_msg void OnBnClickedOk();
    // Name for User Input
    // Name for User Input

```

```

        CEdit m_ctrl_RName;
        // Picture Path for User Input
        CEdit m_ctrl_Rpic_path;
        CEdit m_ctrl_Guideline;
        virtual BOOL OnInitDialog();
};

```

(SELRes.cpp)

```

// SELRes.cpp: 구현 파일
//

#include "pch.h"
#include "RFID.h"
#include "afxdialogex.h"
#include "SELRes.h"

extern CString db_select_result;
// SELRes 대화 상자

IMPLEMENT_DYNAMIC(SELRes, CDialog)

SELRes::SELRes(CWnd* pParent /*=nullptr*/)
    : CDialog(IDD_SELECT_RESULT, pParent)
{

}

SELRes::~SELRes()
{
}

void SELRes::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_EDIT1, m_ctrl_SR);
}

BEGIN_MESSAGE_MAP(SELRes, CDialog)
END_MESSAGE_MAP()

```

```
// SELRes 메시지 처리기
```

```
BOOL SELRes::OnInitDialog()
{
    CDialog::OnInitDialog();
    m_ctrl_SR.SetWindowTextA(db_select_result);

    // TODO: 여기에 추가 초기화 작업을 추가합니다.

    return TRUE; // return TRUE unless you set the focus to a control
    // 예외: OCX 속성 페이지는 FALSE를 반환해야 합니다.
}
```

(SELRes.h)

```
#pragma once
#include "afxdialogex.h"
```

```
// SELRes 대화 상자
```

```
class SELRes : public CDialog
{
    DECLARE_DYNAMIC(SELRes)

public:
    SELRes(CWnd* pParent = nullptr);    // 표준 생성자입니다.
    virtual ~SELRes();

    // 대화 상자 데이터입니다.
#ifdef AFX_DESIGN_TIME
        enum { IDD = IDD_SELECT_RESULT };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 지원입
    니다.
```

```
        DECLARE_MESSAGE_MAP()
public:
        virtual BOOL OnInitDialog();
        CEdit m_ctrl_SR;
};
```