**DataSocket.h**

#pragma once

#define UM\_RECEIVE\_MESSAGE (WM\_USER+101)

#define UM\_CLOSE\_MESSAGE (WM\_USER+102)

// CDataSocket 명령 대상입니다.

class CDataSocket : public CSocket

{

public:

CDataSocket();

virtual ~CDataSocket();

virtual void OnReceive(int nErrorCode);

virtual void OnClose(int nErrorCode);

};

**DataSocket.cpp**

// DataSocket.cpp : 구현 파일입니다.

//

#include "stdafx.h"

#include "KeyboardForAndroidMsg.h"

#include "DataSocket.h"

// CDataSocket

CDataSocket::CDataSocket()

{

}

CDataSocket::~CDataSocket()

{

}

// CDataSocket 멤버 함수

void CDataSocket::OnReceive(int nErrorCode)

{

// TODO: 여기에 특수화된 코드를 추가 및/또는 기본 클래스를 호출합니다.

(\*AfxGetMainWnd()).SendMessage(UM\_RECEIVE\_MESSAGE, (WPARAM)this);

CSocket::OnReceive(nErrorCode);

}

void CDataSocket::OnClose(int nErrorCode)

{

// TODO: 여기에 특수화된 코드를 추가 및/또는 기본 클래스를 호출합니다.

(\*AfxGetMainWnd()).SendMessage(UM\_CLOSE\_MESSAGE, (WPARAM)this);

CSocket::OnClose(nErrorCode);

}

**KeyboardForAndroidMsgDlg.h**

// KeyboardForAndroidMsgDlg.h : 헤더 파일

//

#pragma once

#include "afxwin.h"

#include "afxcmn.h"

#include "DataSocket.h"

#include "atlimage.h"

// CKeyboardForAndroidMsgDlg 대화 상자

class CKeyboardForAndroidMsgDlg : public CDialogEx

{

// 생성입니다.

public:

CKeyboardForAndroidMsgDlg(CWnd\* pParent = NULL); // 표준 생성자입니다.

CDataSocket dataSocket;

CString m\_strServerAddress;

// 대화 상자 데이터입니다.

enum { IDD = IDD\_KEYBOARDFORANDROIDMSG\_DIALOG };

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 지원입니다.

// 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();

afx\_msg LRESULT OnCloseSocket(WPARAM wParam, LPARAM lParam);

afx\_msg LRESULT OnReceiveData(WPARAM wParam, LPARAM lParam);

void AddMessage(CString strMsg);

DECLARE\_MESSAGE\_MAP()

public:

CString m\_strListData;

CString m\_strSendData;

CString m\_strPort;

CEdit m\_edListData;

CIPAddressCtrl m\_ipServerAddress;

afx\_msg void OnBnClickedButtonConnect();

afx\_msg void OnBnClickedButtonStop();

afx\_msg void OnBnClickedButtonSend();

virtual BOOL PreTranslateMessage(MSG\* pMsg);

CImage img;

CString AnsiToUTF8RetCString(CString inputStr);

CRect imgViewerRect;

CStatic imgViewer;

afx\_msg void OnBnClickedUp();

afx\_msg void OnBnClickedLeft();

afx\_msg void OnBnClickedDown();

afx\_msg void OnBnClickedRight();

afx\_msg void OnBnClickedEnter();

afx\_msg void OnCbnSelchangeOpenpack();

CComboBox selOpenPack;

afx\_msg void OnBnClickedBack();

void errorCheck(int ret);

};

**KeyboardForAndroidMsgDlg.cpp**

// KeyboardForAndroidMsgDlg.cpp : 구현 파일

//

#include "stdafx.h"

#include "KeyboardForAndroidMsg.h"

#include "KeyboardForAndroidMsgDlg.h"

#include "afxdialogex.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

// 응용 프로그램 정보에 사용되는 CAboutDlg 대화 상자입니다.

class CAboutDlg : public CDialogEx

{

public:

CAboutDlg();

// 대화 상자 데이터입니다.

enum { IDD = IDD\_ABOUTBOX };

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 지원입니다.

// 구현입니다.

protected:

DECLARE\_MESSAGE\_MAP()

};

CAboutDlg::CAboutDlg() : CDialogEx(CAboutDlg::IDD)

{

}

void CAboutDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

}

BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialogEx)

END\_MESSAGE\_MAP()

// CKeyboardForAndroidMsgDlg 대화 상자

CKeyboardForAndroidMsgDlg::CKeyboardForAndroidMsgDlg(CWnd\* pParent /\*=NULL\*/)

: CDialogEx(CKeyboardForAndroidMsgDlg::IDD, pParent)

, m\_strListData(\_T(""))

, m\_strSendData(\_T(""))

, m\_strPort(\_T(""))

{

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CKeyboardForAndroidMsgDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Text(pDX, IDC\_EDIT\_LIST, m\_strListData);

DDX\_Text(pDX, IDC\_EDIT\_SEND\_DATA, m\_strSendData);

DDX\_Text(pDX, IDC\_EDIT\_PORT, m\_strPort);

DDX\_Control(pDX, IDC\_EDIT\_LIST, m\_edListData);

DDX\_Control(pDX, IDC\_IPADDRESS, m\_ipServerAddress);

DDX\_Control(pDX, IDC\_PICVIEW, imgViewer);

DDX\_Control(pDX, IDC\_OPENPACK, selOpenPack);

}

BEGIN\_MESSAGE\_MAP(CKeyboardForAndroidMsgDlg, CDialogEx)

ON\_WM\_SYSCOMMAND()

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

ON\_BN\_CLICKED(IDC\_BUTTON\_CONNECT, &CKeyboardForAndroidMsgDlg::OnBnClickedButtonConnect)

ON\_BN\_CLICKED(IDC\_BUTTON\_STOP, &CKeyboardForAndroidMsgDlg::OnBnClickedButtonStop)

ON\_BN\_CLICKED(IDC\_BUTTON\_SEND, &CKeyboardForAndroidMsgDlg::OnBnClickedButtonSend)

ON\_BN\_CLICKED(IDC\_UP, &CKeyboardForAndroidMsgDlg::OnBnClickedUp)

ON\_BN\_CLICKED(IDC\_LEFT, &CKeyboardForAndroidMsgDlg::OnBnClickedLeft)

ON\_BN\_CLICKED(IDC\_DOWN, &CKeyboardForAndroidMsgDlg::OnBnClickedDown)

ON\_BN\_CLICKED(IDC\_RIGHT, &CKeyboardForAndroidMsgDlg::OnBnClickedRight)

ON\_BN\_CLICKED(IDC\_ENTER, &CKeyboardForAndroidMsgDlg::OnBnClickedEnter)

ON\_CBN\_SELCHANGE(IDC\_OPENPACK, &CKeyboardForAndroidMsgDlg::OnCbnSelchangeOpenpack)

ON\_BN\_CLICKED(IDC\_BACK, &CKeyboardForAndroidMsgDlg::OnBnClickedBack)

END\_MESSAGE\_MAP()

// CKeyboardForAndroidMsgDlg 메시지 처리기

BOOL CKeyboardForAndroidMsgDlg::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 != NULL)

{

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 아닐 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.

// 픽처컨트롤의 크기를 tmpRect에 저장

CRect tmpRect;

GetDlgItem(IDC\_PICVIEW)->MoveWindow(10, 10, 300, 370);

GetDlgItem(IDC\_PICVIEW)->GetClientRect(tmpRect);

// 픽처컨트롤이 tmpRect의 크기로 그리게 되면 0,0부터 설정되므로 x, y에 10씩 추가

imgViewerRect.SetRect(tmpRect.left+10, tmpRect.top+10, tmpRect.right+10, tmpRect.bottom+10);

// 카톡 이미지를 로드, 같은 폴더에 이미지가 없으면 이미지 로드가 안됨

img.Load(CString("katalk.jpg"));

Invalidate();

// 메시지를 IDC\_EDIT\_LIST에 출력

AddMessage("기본 ip : 192.168.0.11");

AddMessage("기본 포트 3600");

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

}

void CKeyboardForAndroidMsgDlg::OnSysCommand(UINT nID, LPARAM lParam)

{

if ((nID & 0xFFF0) == 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 응용 프로그램의 경우에는

// 프레임워크에서 이 작업을 자동으로 수행합니다.

void CKeyboardForAndroidMsgDlg::OnPaint()

{

if (IsIconic())

{

CPaintDC dc(this); // 그리기를 위한 디바이스 컨텍스트입니다.

SendMessage(WM\_ICONERASEBKGND, 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

{

// 카톡 이미지를 픽처컨트롤에 출력

CPaintDC dc(this);

img.Draw(dc.m\_hDC, imgViewerRect);

CDialogEx::OnPaint();

}

}

// 사용자가 최소화된 창을 끄는 동안에 커서가 표시되도록 시스템에서

// 이 함수를 호출합니다.

HCURSOR CKeyboardForAndroidMsgDlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

void CKeyboardForAndroidMsgDlg::OnBnClickedButtonConnect()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

UpdateData();

int port;

// 포트주소를 입력했으면 입력된 포트로 연결

if(m\_strPort != "")

{

port = atoi((LPCSTR)m\_strPort);

}

// 포트주소가 입력되지 않았으면 기본 포트 3600을 사용해서 연결

else

{

port = 3600;

}

// IP주소를 입력했으면 입력된 IP주소로 연결

if(m\_strServerAddress != "")

{

m\_ipServerAddress.GetWindowText(m\_strServerAddress);

}

// IP주소를 입력하지 않으면 기본 IP 192.168.0.11로 연결

else

{

m\_strServerAddress = "192.168.0.11";

}

// SOCKET m\_hSocket는 CAsyncSocket 클래스에 있는 변수

if(dataSocket.m\_hSocket != INVALID\_SOCKET)

{

AddMessage("이미 서버에 접속 되어있습니다.");

}

if(!dataSocket.Create())

{

AddMessage("소켓 생성 실패");

}

// IP와 포트로 연결에 실패하는 경우

if(!dataSocket.Connect(m\_strServerAddress,port))

{

dataSocket.Close();

AddMessage("서버 접속 실패");

}

// 연결 성공시 연결된 IP와 포트번호 출력

else

{

CString strPt;

strPt.Format("포트번호 %d", port);

AddMessage(strPt);

AddMessage(m\_strServerAddress);

AddMessage("서버에 접속 됐습니다.");

// 접속끊기 버튼 활성화, 서버접속 버튼 비활성화

GetDlgItem(IDC\_BUTTON\_STOP)->EnableWindow(TRUE);

GetDlgItem(IDC\_BUTTON\_CONNECT)->EnableWindow(FALSE);

// 보내기 버튼 활성화, 메세지 입력창에 포커스를 줌

GetDlgItem(IDC\_BUTTON\_SEND)->EnableWindow(TRUE);

GetDlgItem(IDC\_EDIT\_SEND\_DATA)->SetFocus();

}

}

void CKeyboardForAndroidMsgDlg::OnBnClickedButtonStop()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

// dataSocket이 유효한 소켓인 경우에 종료 절차를 거침

if(dataSocket.m\_hSocket != INVALID\_SOCKET)

{

dataSocket.Close();

AddMessage("서버와의 접속을 종료합니다.");

// 접속끊기 버튼 비활성화, 서버접속 버튼 활성화, 보내기 버튼 비활성화

GetDlgItem(IDC\_BUTTON\_STOP)->EnableWindow(FALSE);

GetDlgItem(IDC\_BUTTON\_CONNECT)->EnableWindow(TRUE);

GetDlgItem(IDC\_BUTTON\_SEND)->EnableWindow(FALSE);

}

else

{

AddMessage("이미 종료되었습니다.");

}

}

void CKeyboardForAndroidMsgDlg::OnBnClickedButtonSend()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

// 입력한 메세지를 strSend에 할당

UpdateData();

CString strSend;

strSend.Format("%s", m\_strSendData);

// Ansi를 UTF8로 변경하고 CString으로 리턴된 것을 전송

dataSocket.Send(AnsiToUTF8RetCString(strSend),

AnsiToUTF8RetCString(strSend).GetLength()+1);

GetDlgItem(IDC\_EDIT\_SEND\_DATA)->SetFocus();

m\_strSendData = "";

UpdateData(FALSE);

}

// 키보드 제어를 미리 가로채서 처리해주는 메소드(재정의함)

BOOL CKeyboardForAndroidMsgDlg::PreTranslateMessage(MSG\* pMsg)

{

// TODO: 여기에 특수화된 코드를 추가 및/또는 기본 클래스를 호출합니다.

// 키가 눌렸을때

if(pMsg->message == WM\_KEYDOWN)

{

// 엔터키가 눌렸으면

if(pMsg->wParam == VK\_RETURN)

{

// 포커스가 메세지 입력창에 있으면

if(GetFocus()->GetDlgCtrlID() == IDC\_EDIT\_SEND\_DATA)

{

// 메세지 전송 수행

OnBnClickedButtonSend();

return TRUE;

}

else

{

// 그냥 엔터만 눌리면 엔터 명령 전송

OnBnClickedEnter();

return TRUE;

}

}

// ESC가 눌리면

else if(pMsg->wParam == VK\_ESCAPE)

{

// 뒤로가기 명령 전송

OnBnClickedBack();

return TRUE;

}

// 방향키가 눌리면 각각의 방향 명령 전송

switch(pMsg->wParam){

case VK\_LEFT:

OnBnClickedLeft();

break;

case VK\_RIGHT:

OnBnClickedRight();

break;

case VK\_UP:

OnBnClickedUp();

break;

case VK\_DOWN:

OnBnClickedDown();

break;

}

}

return CDialogEx::PreTranslateMessage(pMsg);

}

// 왼쪽 에디트 컨트롤에 메시지를 출력해주는 메소드

void CKeyboardForAndroidMsgDlg::AddMessage(CString strMsg)

{

UpdateData();

// 이전 메세지가 없을때

if(m\_strListData == "")

m\_strListData = strMsg;

// 그전에 메세지가 있을때

else

m\_strListData = m\_strListData + "\r\n" + strMsg;

UpdateData(FALSE);

m\_edListData.LineScroll(m\_edListData.GetLineCount());

}

// 서버로부터 온 메세지를 받아서 에디드 컨트롤에 출력

LRESULT CKeyboardForAndroidMsgDlg::OnReceiveData(WPARAM wParam, LPARAM lParam)

{

char Rcvdata[1024];

CDataSocket\* pDataSocket = (CDataSocket\*)wParam;

pDataSocket->Receive(Rcvdata, sizeof(Rcvdata));

CString strMsg = Rcvdata;

AddMessage(strMsg);

return 0;

}

// 서버와의 연결 종료 처리

LRESULT CKeyboardForAndroidMsgDlg::OnCloseSocket(WPARAM wParam, LPARAM lParam)

{

dataSocket.Close();

AddMessage("서버가 연결을 종료 했습니다.");

// 접속끊기 버튼 비활성화, 서버접속 버튼 활성화, 보내기 버튼 비활성화

GetDlgItem(IDC\_BUTTON\_STOP)->EnableWindow(FALSE);

GetDlgItem(IDC\_BUTTON\_CONNECT)->EnableWindow(TRUE);

GetDlgItem(IDC\_BUTTON\_SEND)->EnableWindow(FALSE);

return 0;

}

// Ansi를 UTF8 형식으로 변환하여 CString으로 리턴

CString CKeyboardForAndroidMsgDlg::AnsiToUTF8RetCString(CString inputStr)

{

WCHAR szUnicode[1024];

char szUTF8char[1024];

CString strConvert;

char\* szSrc = (LPSTR)(LPCTSTR)inputStr;

char szRetValue[1024] = "";

int unicodeSize = MultiByteToWideChar(CP\_ACP, 0,

szSrc, (int)strlen(szSrc),

szUnicode, sizeof(szUnicode));

int UTF8CodeSize = WideCharToMultiByte(CP\_UTF8, 0,

szUnicode, unicodeSize, szUTF8char,

sizeof(szUTF8char), NULL, NULL);

memcpy(szRetValue, szUTF8char, UTF8CodeSize);

strConvert = szRetValue;

return strConvert;

}

// ↑ 눌렀을때 처리

void CKeyboardForAndroidMsgDlg::OnBnClickedUp()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

UpdateData();

CString strUp;

strUp.Format("%s", "\\\\CONTROL\_U");

dataSocket.Send(strUp, strUp.GetLength()+1);

GetDlgItem(IDC\_IPADDRESS)->SetFocus();

GetDlgItem(IDC\_IPADDRESS)->SendMessage(WM\_KILLFOCUS, NULL);

GetDlgItem(IDC\_UP)->SendMessage(WM\_KILLFOCUS, NULL);

}

// ← 눌렀을때 처리

void CKeyboardForAndroidMsgDlg::OnBnClickedLeft()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

CString strLeft;

strLeft.Format("%s", "\\\\CONTROL\_L");

dataSocket.Send(strLeft, strLeft.GetLength()+1);

GetDlgItem(IDC\_IPADDRESS)->SetFocus();

GetDlgItem(IDC\_IPADDRESS)->SendMessage(WM\_KILLFOCUS, NULL);

GetDlgItem(IDC\_EDIT\_SEND\_DATA)->SendMessage(WM\_KILLFOCUS, NULL);

}

// ↓ 눌렀을때 처리

void CKeyboardForAndroidMsgDlg::OnBnClickedDown()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

CString strDown;

strDown.Format("%s", "\\\\CONTROL\_D");

dataSocket.Send(strDown, strDown.GetLength()+1);

GetDlgItem(IDC\_IPADDRESS)->SetFocus();

GetDlgItem(IDC\_IPADDRESS)->SendMessage(WM\_KILLFOCUS, NULL);

GetDlgItem(IDC\_DOWN)->SendMessage(WM\_KILLFOCUS, NULL);

}

// → 눌렀을때 처리

void CKeyboardForAndroidMsgDlg::OnBnClickedRight()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

CString strRight;

strRight.Format("%s", "\\\\CONTROL\_R");

dataSocket.Send(strRight, strRight.GetLength()+1);

GetDlgItem(IDC\_IPADDRESS)->SetFocus();

GetDlgItem(IDC\_IPADDRESS)->SendMessage(WM\_KILLFOCUS, NULL);

GetDlgItem(IDC\_RIGHT)->SendMessage(WM\_KILLFOCUS, NULL);

}

// 엔터키 명령을 전송할때 처리

void CKeyboardForAndroidMsgDlg::OnBnClickedEnter()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

CString strEnter;

strEnter.Format("%s", "\\\\CONTROL\_E");

dataSocket.Send(strEnter, strEnter.GetLength()+1);

GetDlgItem(IDC\_IPADDRESS)->SetFocus();

GetDlgItem(IDC\_IPADDRESS)->SendMessage(WM\_KILLFOCUS, NULL);

GetDlgItem(IDC\_ENTER)->SendMessage(WM\_KILLFOCUS, NULL);

}

// ESC 명령을 전송할때 처리

void CKeyboardForAndroidMsgDlg::OnBnClickedBack()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

CString strBack;

strBack.Format("%s", "\\\\CONTROL\_B");

dataSocket.Send(strBack, strBack.GetLength()+1);

GetDlgItem(IDC\_IPADDRESS)->SetFocus();

GetDlgItem(IDC\_IPADDRESS)->SendMessage(WM\_KILLFOCUS, NULL);

GetDlgItem(IDC\_BACK)->SendMessage(WM\_KILLFOCUS, NULL);

}

// 응용프로그램을 선택할때 처리하는 부분

void CKeyboardForAndroidMsgDlg::OnCbnSelchangeOpenpack()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

int nIndex = selOpenPack.GetCurSel();

if(nIndex == -1){

return;

}

CString str, cmd;

selOpenPack.GetLBText(nIndex, str);

// 카톡을 선택하면

if(str == "카톡")

{

// 카톡을 실행하라는 명령을 전송

cmd.Format("%s", "\\\\OPENPACK\_com.kakao.talk");

dataSocket.Send(cmd, cmd.GetLength()+1);

GetDlgItem(IDC\_EDIT\_SEND\_DATA)->SetFocus();

UpdateData(FALSE);

AddMessage("카카오톡 실행");

}

// 라인을 선택하면

else if(str == "라인")

{

// 라인을 실행하라는 명령을 전송

cmd.Format("%s", "\\\\OPENPACK\_jp.naver.line.android");

dataSocket.Send(cmd, cmd.GetLength()+1);

GetDlgItem(IDC\_EDIT\_SEND\_DATA)->SetFocus();

UpdateData(FALSE);

AddMessage("라인 실행");

}

}