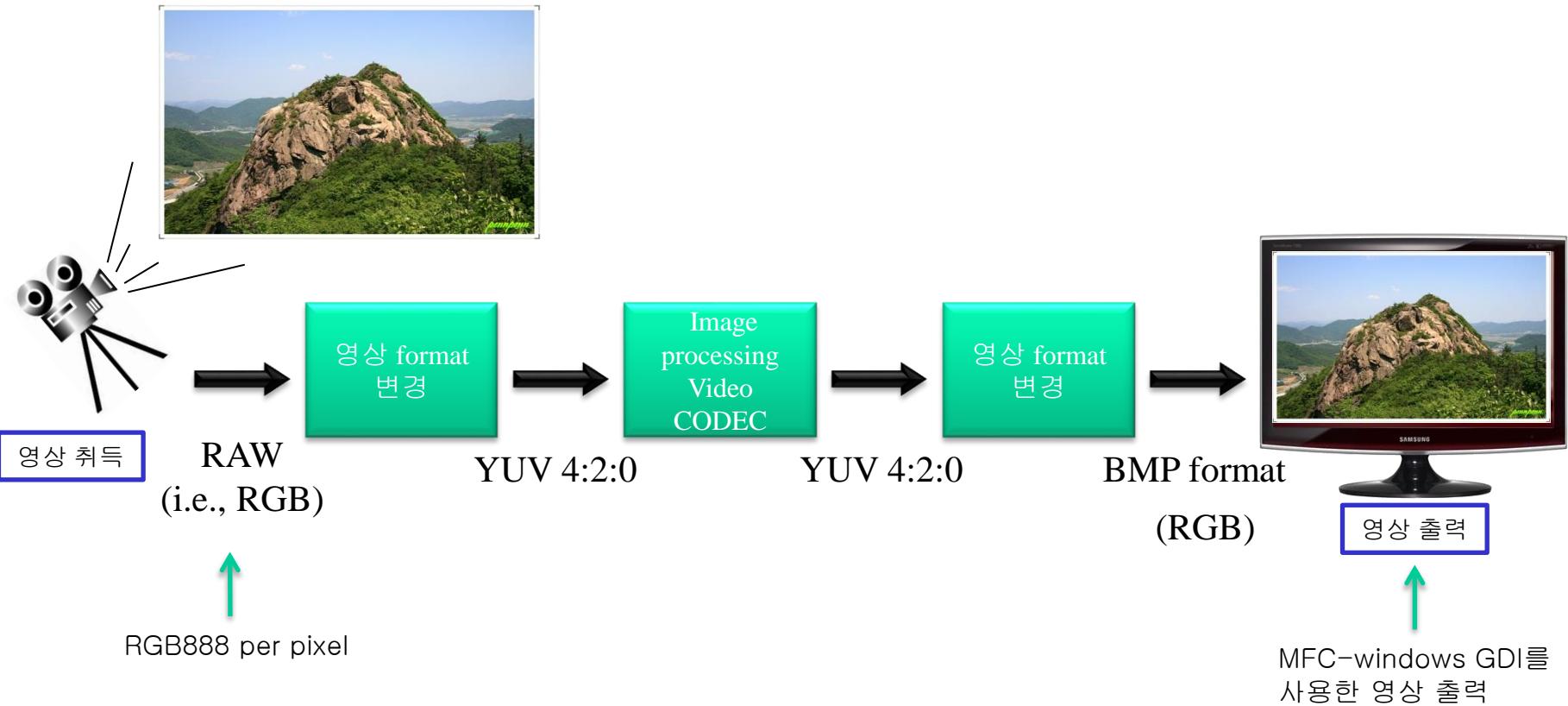


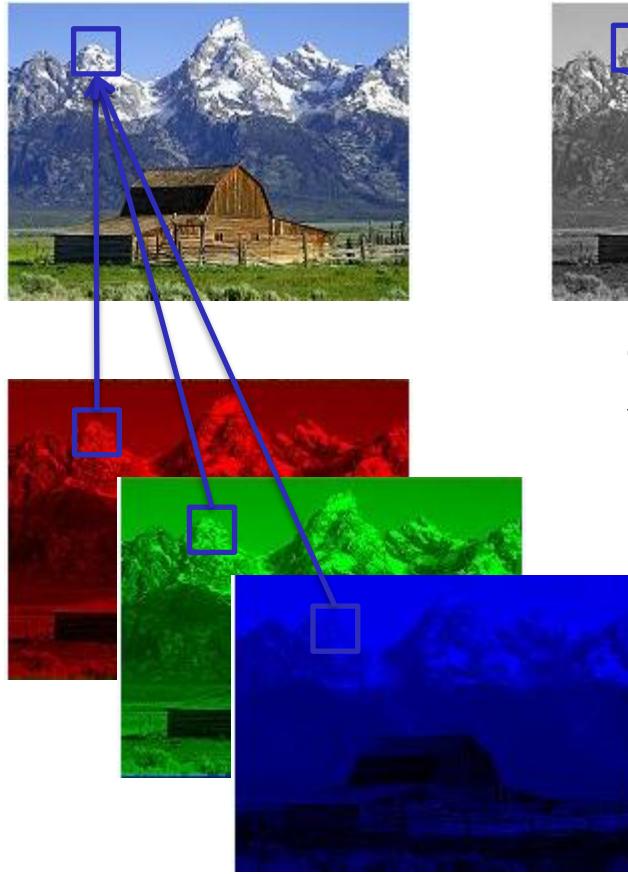
CONTENTS

- 영상의 취득과 출력
- Image format
 - RGB & BMP format
 - YUV format
- Raw 이미지 뷰어 만들기
 - 오픈소스 뷰어 pYUV 소개
 - 파일 불러오기
 - BMP 변환
 - 출력
- 결과

영상의 취득과 출력



RGB format



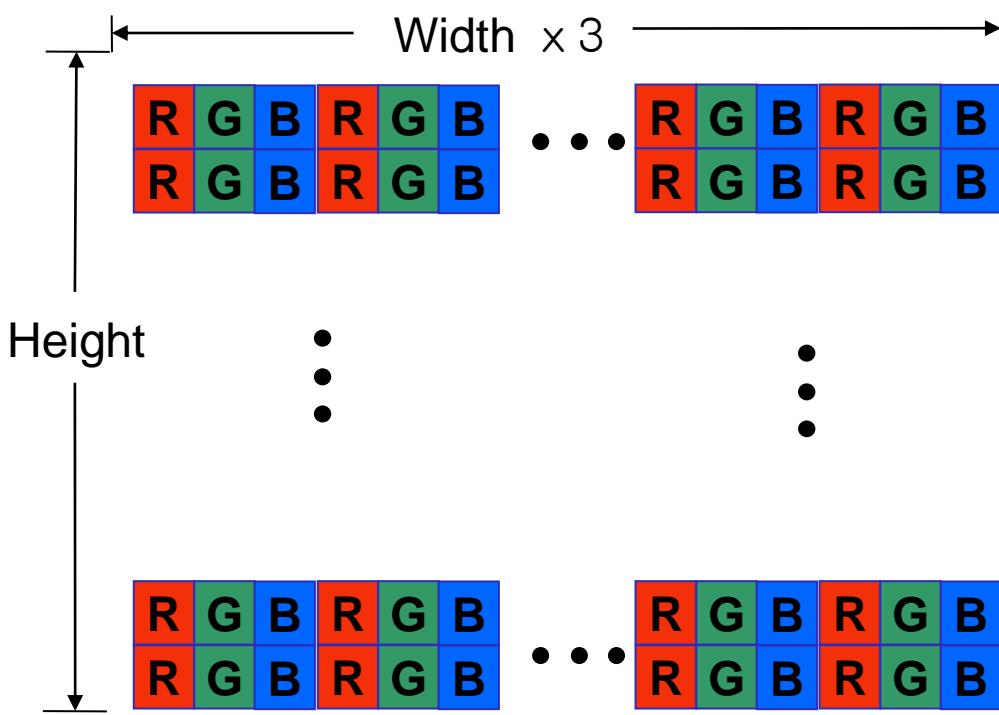
Gray Image
Y(Luma)



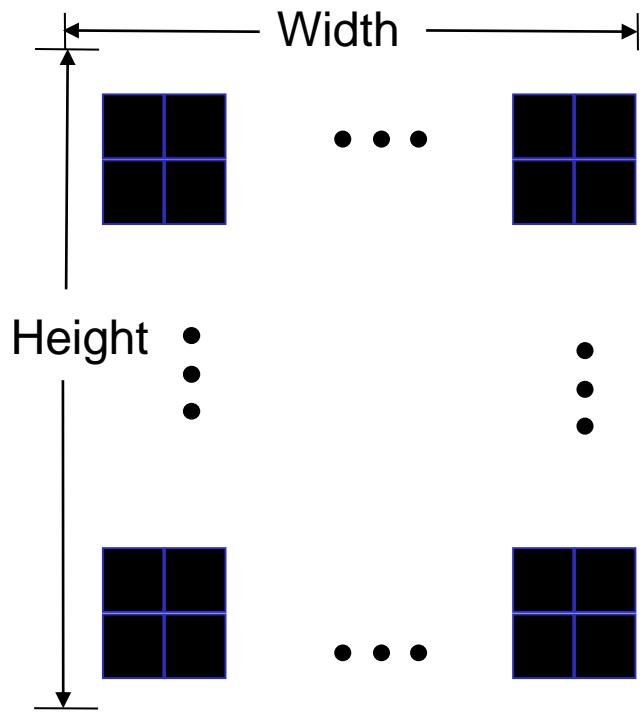
- 영상의 구성
 - 컬러 영상 : 각 픽셀은 Red, Green, Blue의 3개의 원소
 - 흑백 영상 : 각 픽셀은 1개의 원소(R,G,B가 같은 값)

Gray & COLOR RAW file format

- COLOR

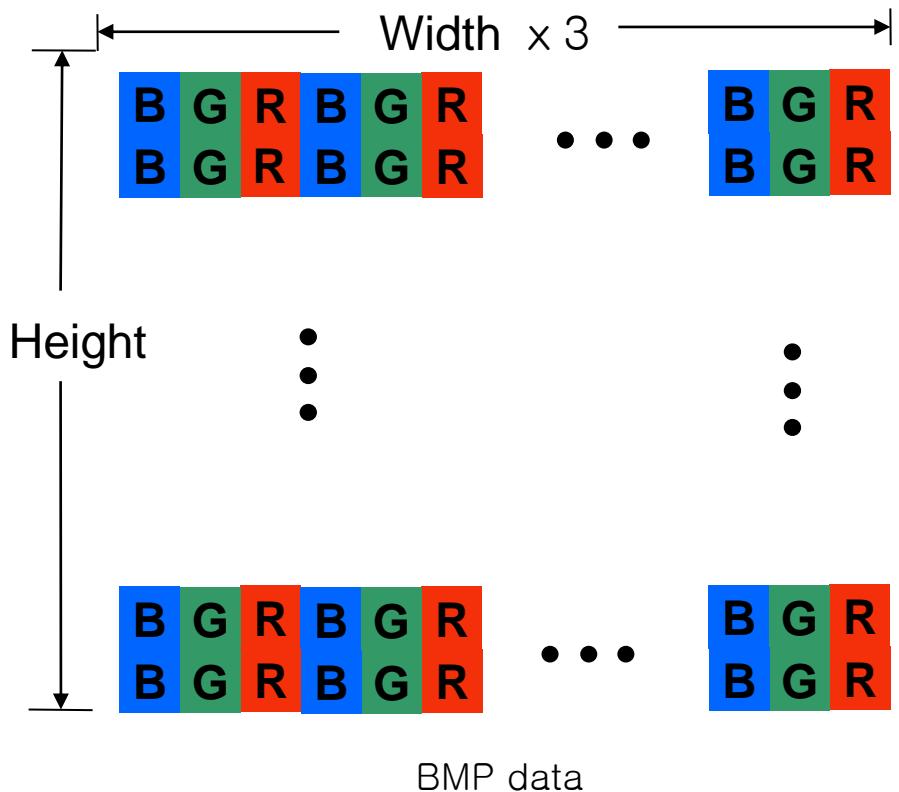
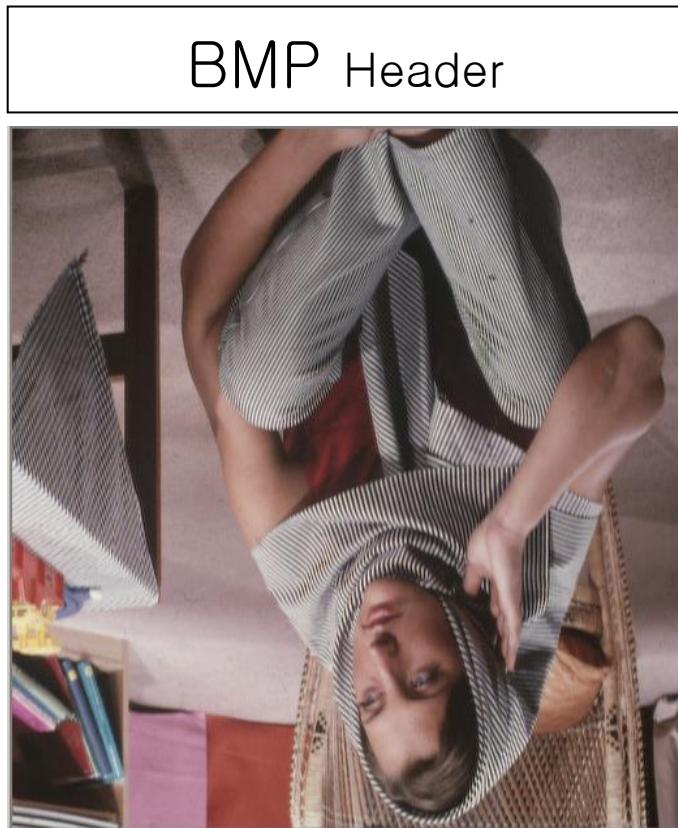


- Gray



BMP file format

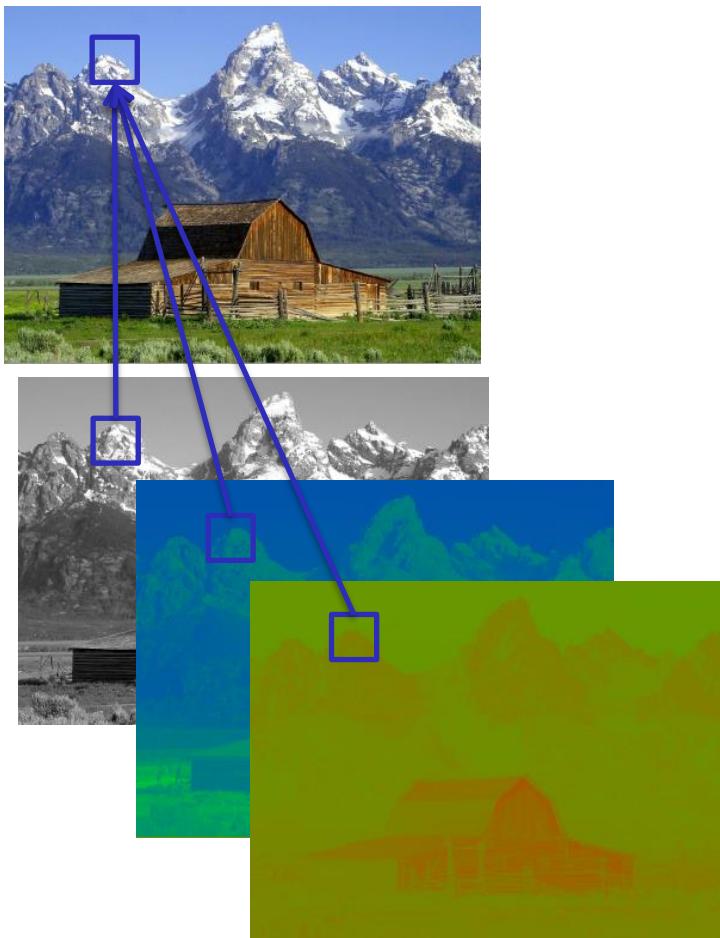
- BMP file format (Header + BMP data)
 - 1 pixel(1-byte)을 BGR(3-bytes)로 표현
 - 메모리상에 영상 **상하 반전** 형태로 저장



YUV format

□ YUV format

➤ 휘도와 색차 성분들로 구성됨

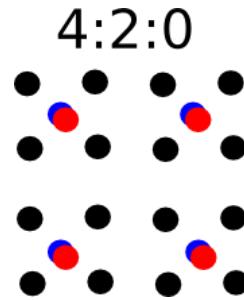


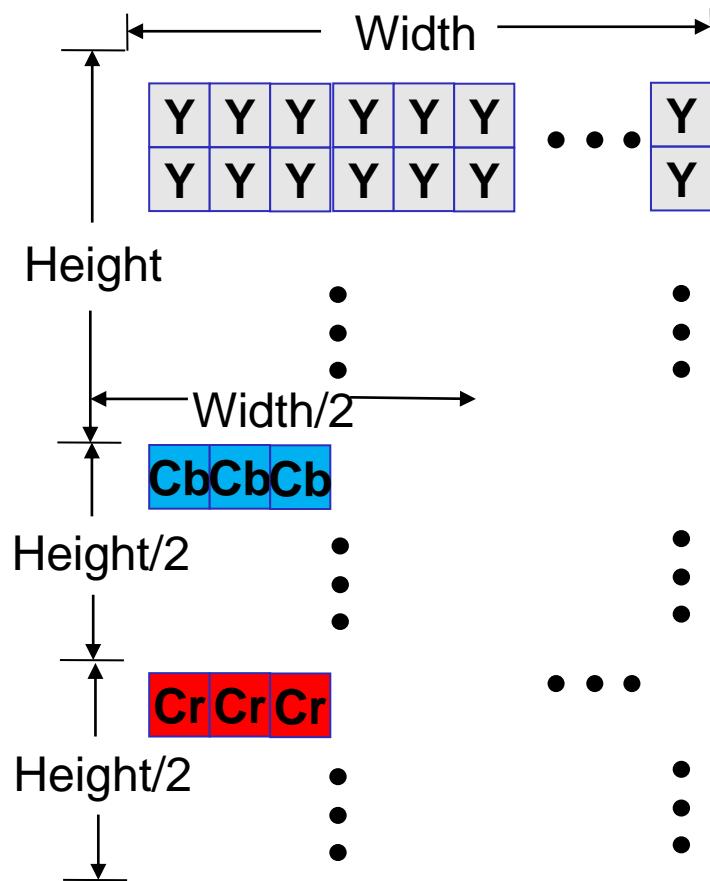
• 영상의 구성

- 휘도(Y) : 밝기
- Cb(U) : chroma blue
- Cr(V) : chroma red

YUV COLOR RAW file format

- COLOR

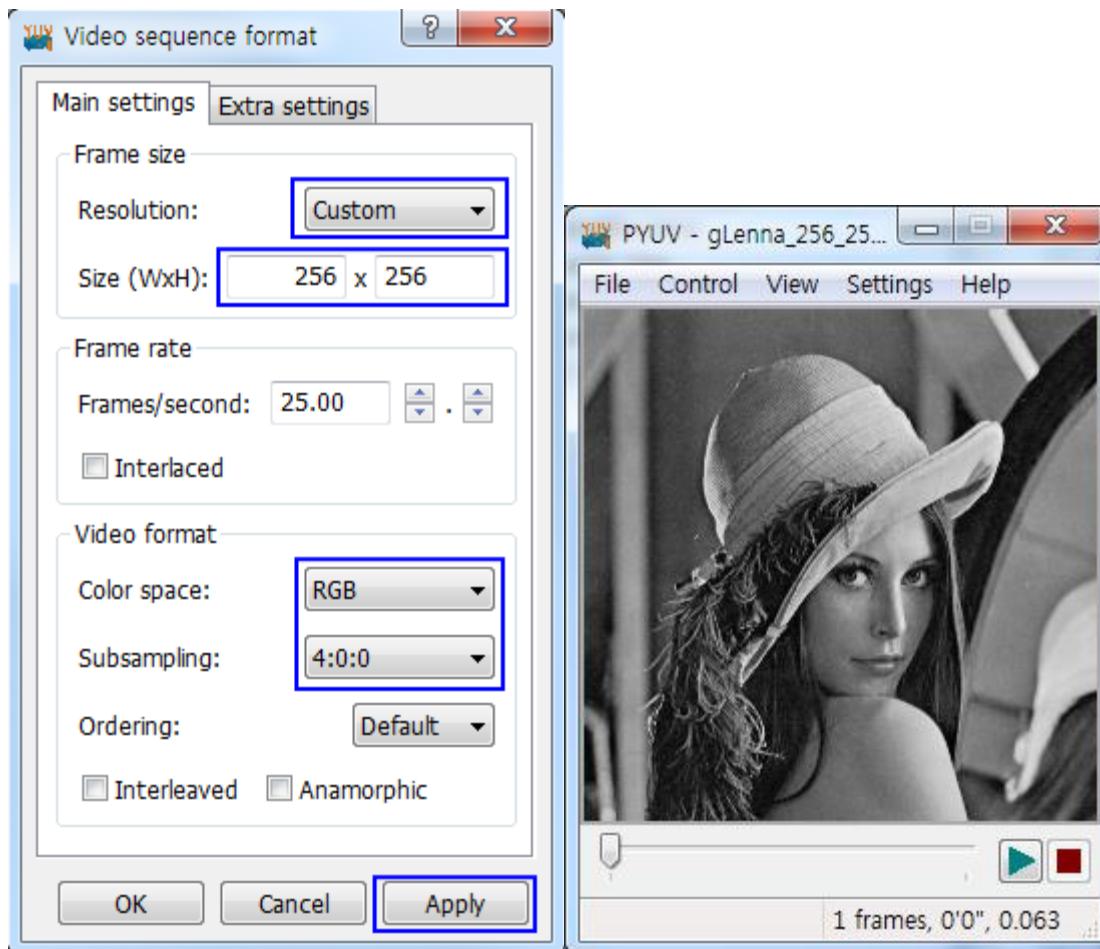




pYUV

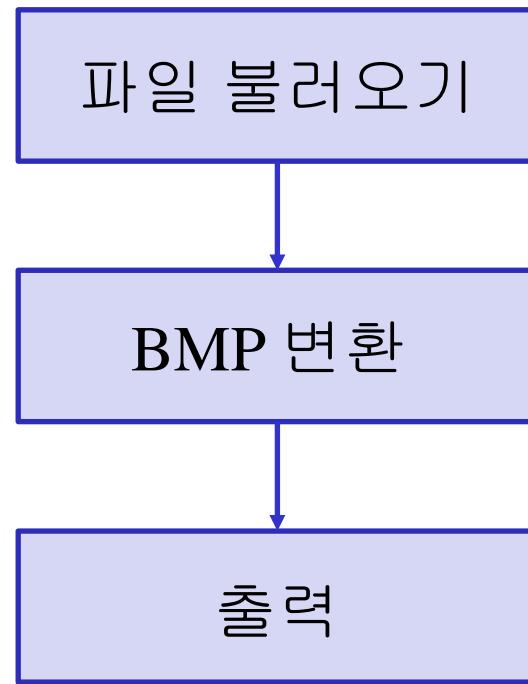
□ 다운로드 경로

➤ <http://dsplab.diei.unipg.it/~baruffa/dvbt/binaries/player/win32/>



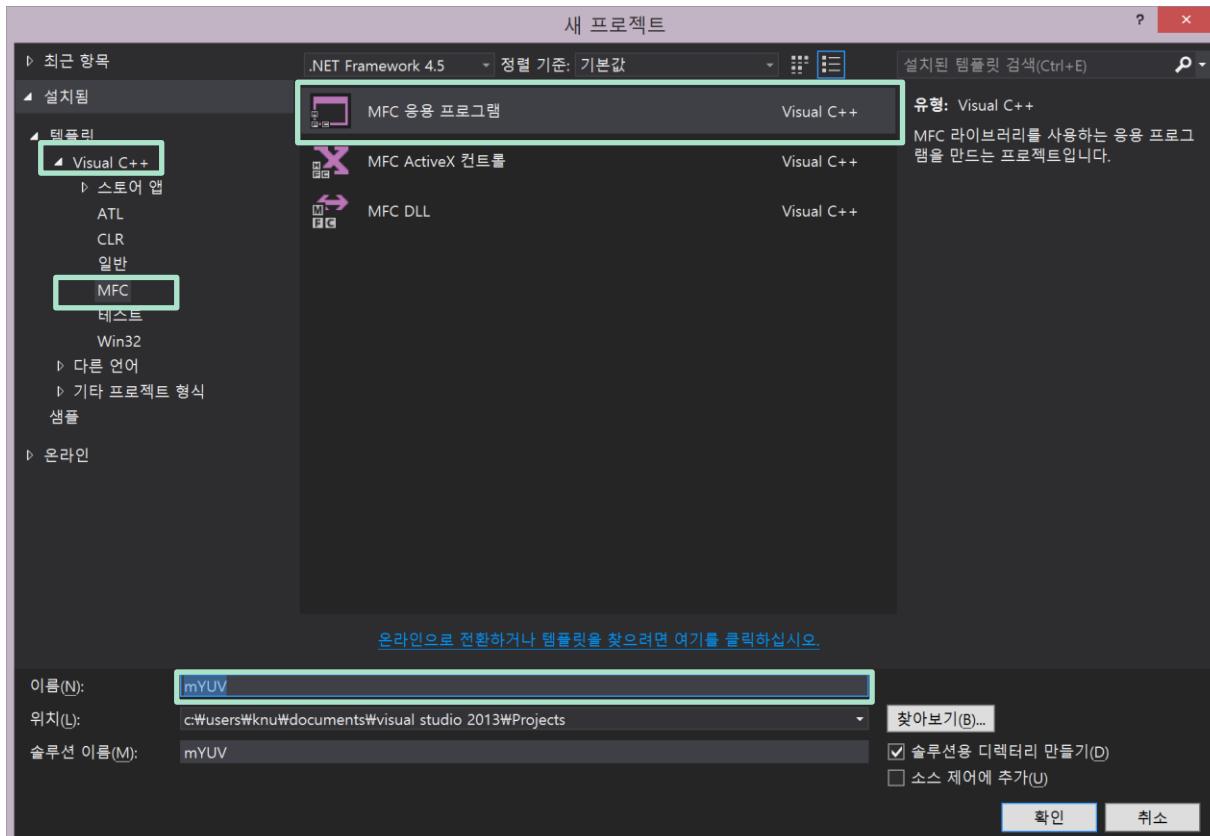
프로그램 순서도

Bitmap.mp4 비디오 참조하면서 볼 것 (MFC)



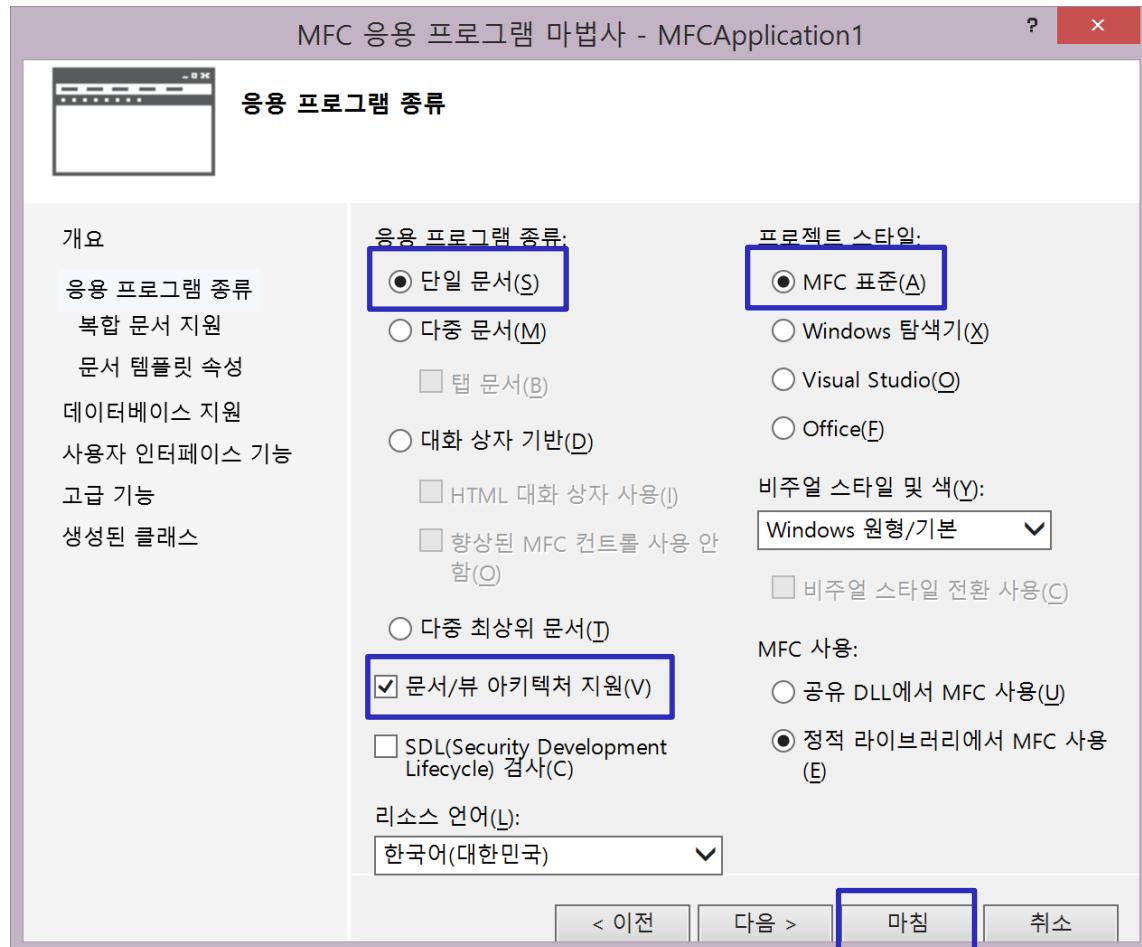
프로젝트 생성

- 1 . Visual Studio 를 킨다.
- 2 . Ctrl + Shift + N을 눌러 새 프로젝트창을 띠운다.
- 3 . Visual C++ → MFC → MFC 응용 프로그램 선택
- 4 . 이름은 "mYUV" 또는 하고싶은 이름으로 작성한다.



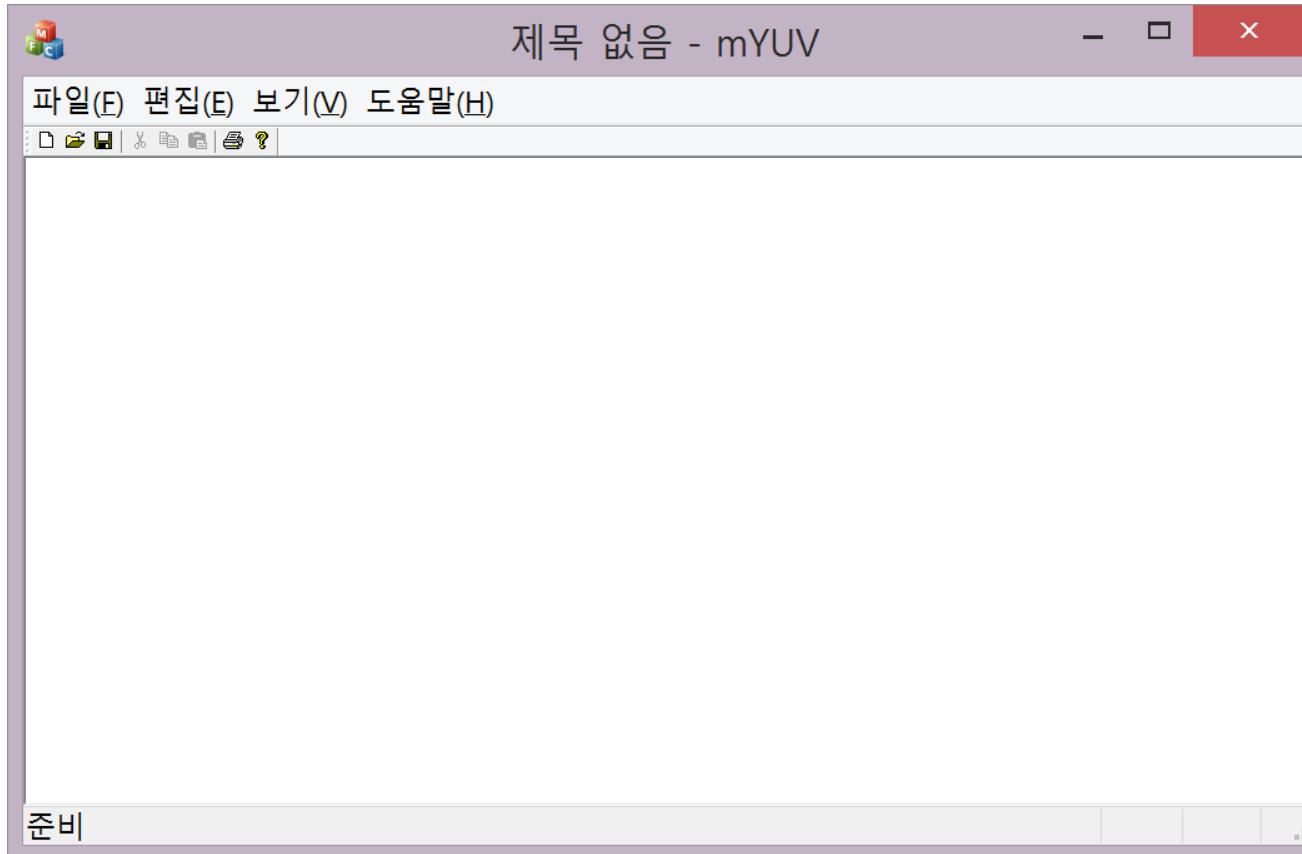
프로젝트 생성

- 단일문서, MFC표준, 문서뷰 아키텍쳐 지원을 선택하고 마친다.



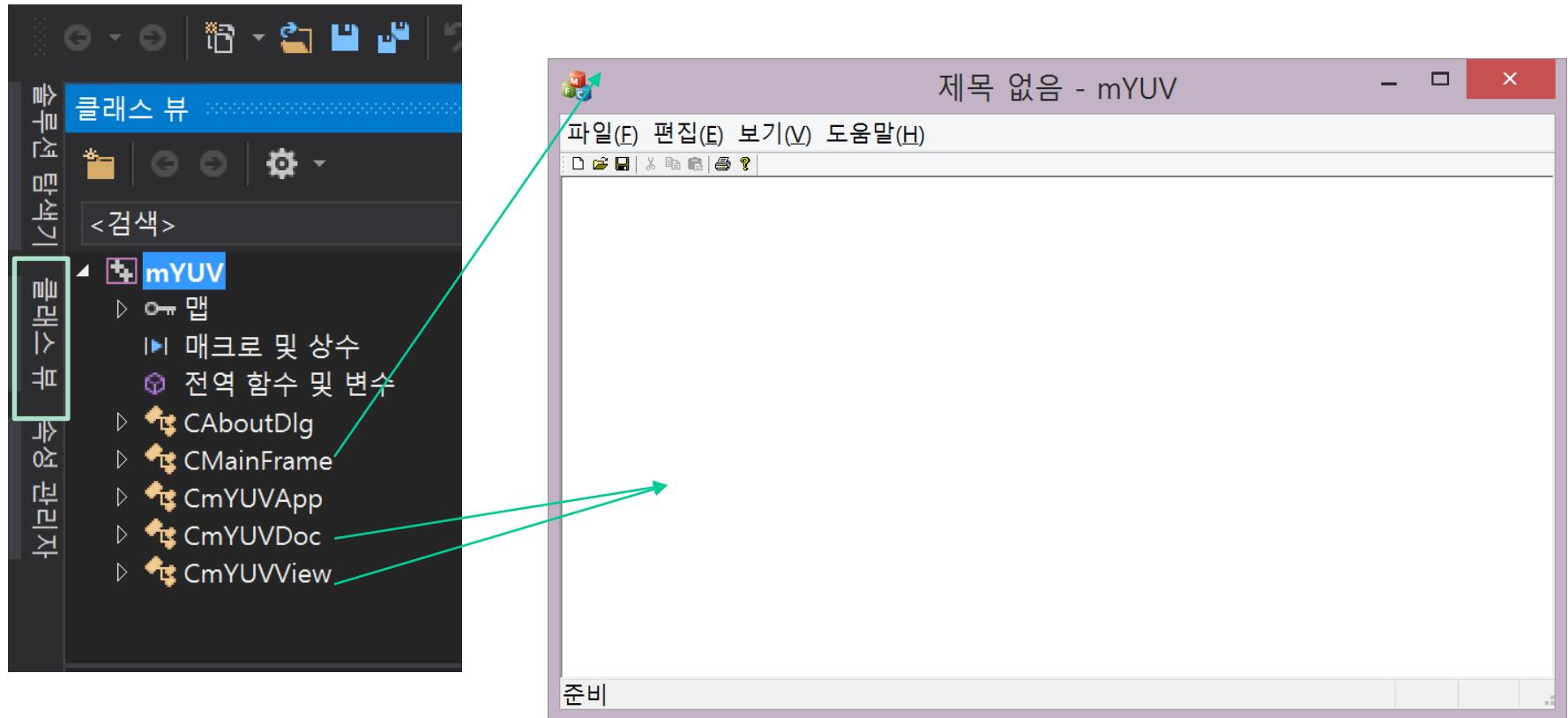
프로젝트 생성

- Ctrl + F5 를 눌러 실행을 해보면 아래와 같은 창이뜬다.



프로젝트 구성

□ 생성된 클래스



프로젝트 구성

□ CMainFrame

- 메뉴, 툴바, 상태창, 제목표시줄 등 뼈대 부분을 기술한 클래스

□ C~App

- 프로그램의 시작 클래스 (최상단 계층)

□ C~Doc

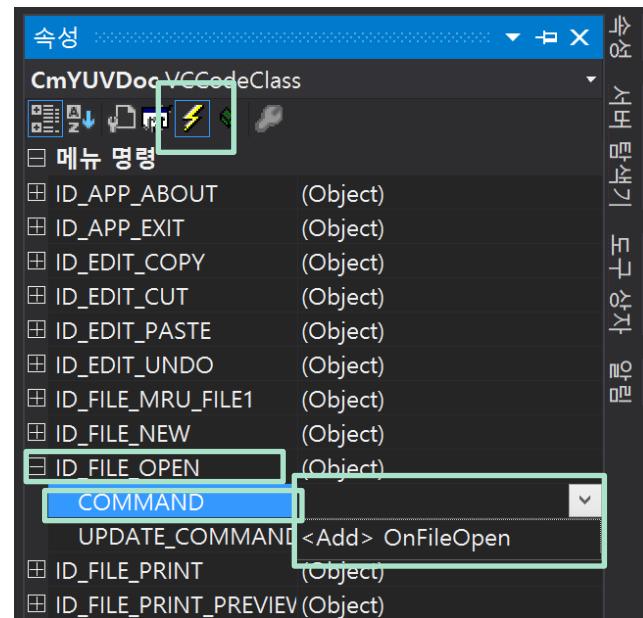
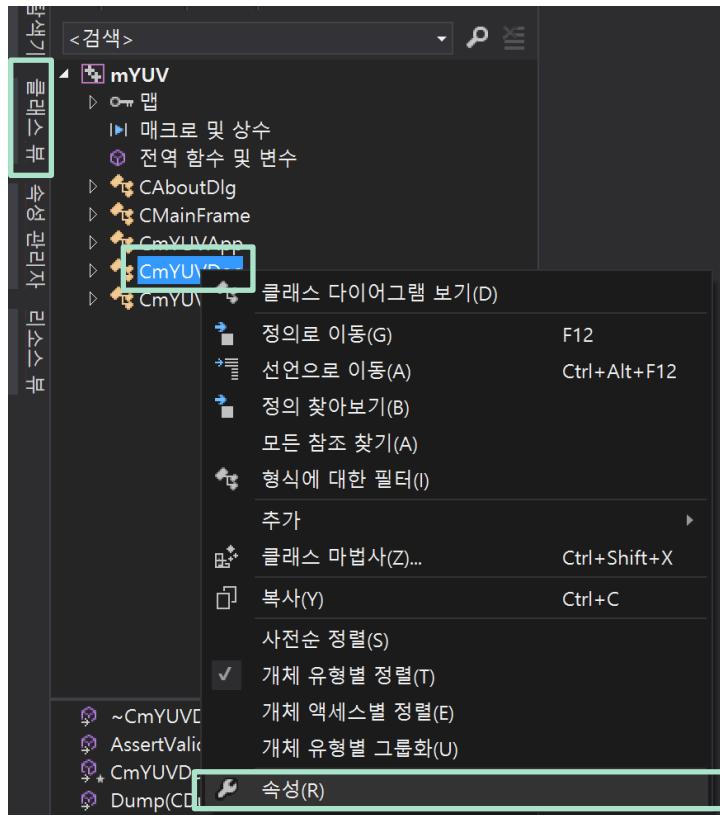
- 데이터를 보관하고, 파일입출력도 하는 클래스

□ C~View

- 데이터를 화면에 그려주는 클래스

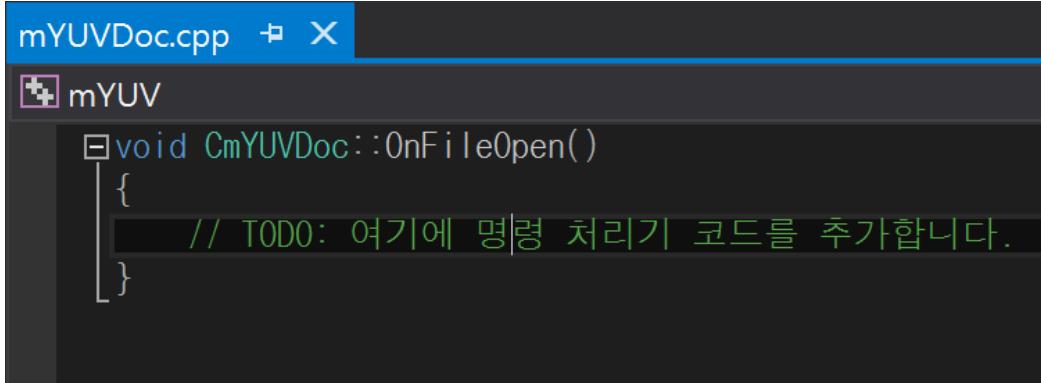
파일 불러오기

- 클래스뷰에서 C~Doc 클래스를 선택후 속성누름.
- 속성창에서 이벤트(번개)를 선택.
- ID_FILE_OPEN 에서 COMMAND의 OnFileOpen 선택



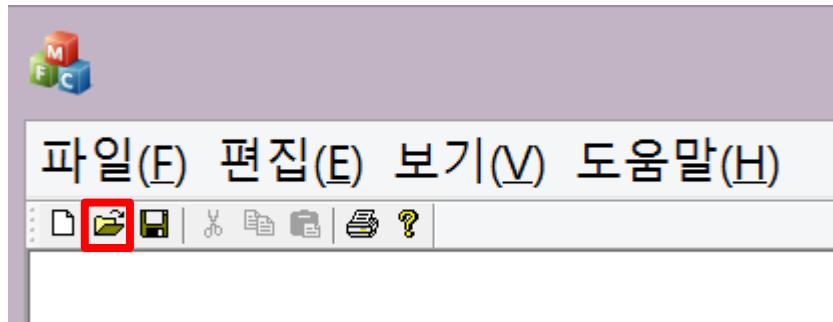
파일 불러오기

- 완료되면 아래와같은 OnFileOpen 함수가 생성된다.



```
mYUVDoc.cpp  ✎ X
mYUV
void CmYUVDoc::OnFileOpen()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
}
```

- OnFileOpen 함수는 **열기**버튼을 누르면 들어옴.

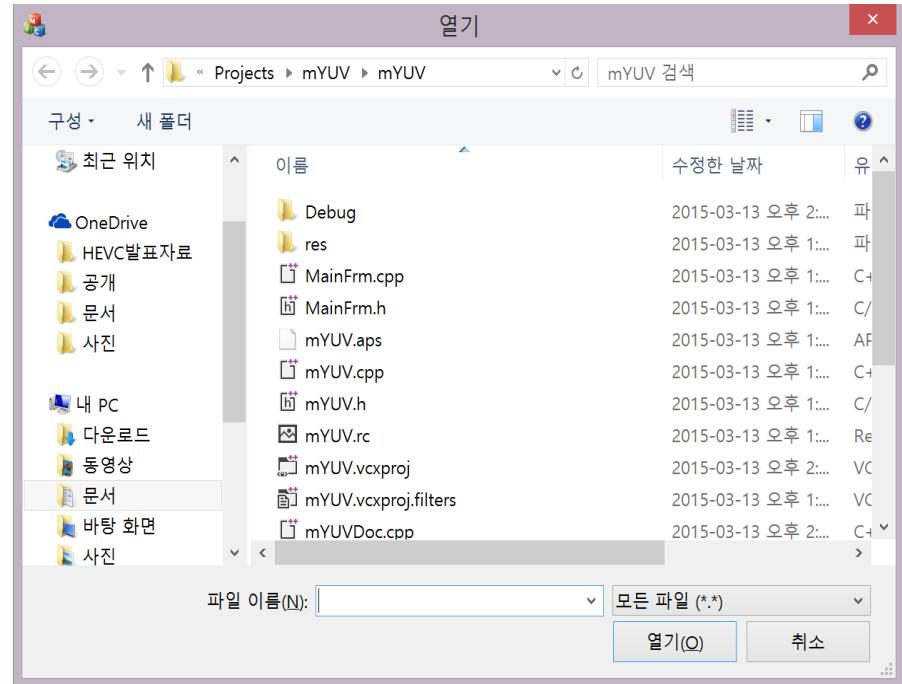


파일 불러오기

- 파일경로를 알아내는 코드를 추가한다.
- CFileDialog 클래스는 열기 대화상자를 생성해준다.

```
void CmYUVDoc::OnFileOpen()
{
    // TODO: 여기에 명령 처리기 코드를 추가
    INT_PTR iDialogResult;
    CFileDialog fileDialog(true);

    iDialogResult = fileDialog.DoModal();
```



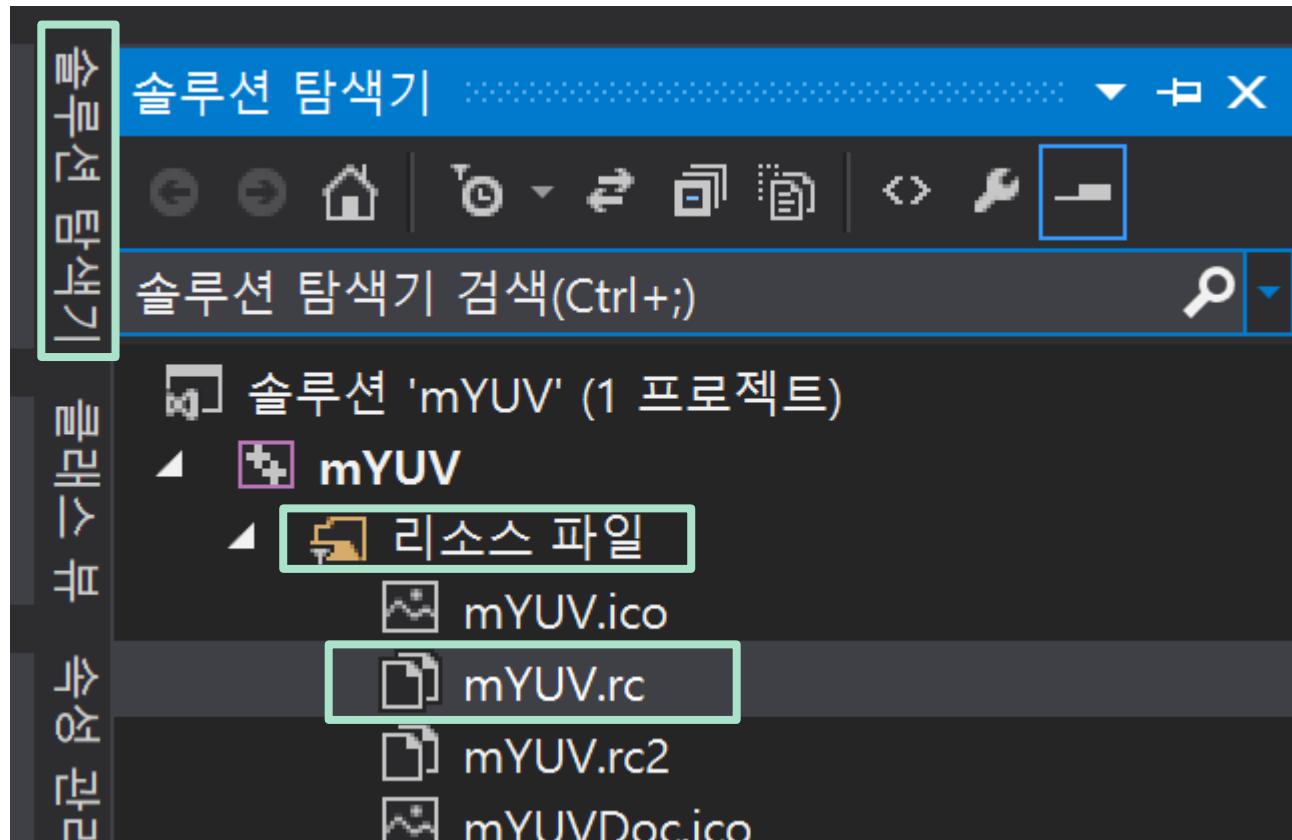
파일 불러오기 - 다이얼로그 생성

- 불러올 Raw파일에는 이미지 너비, 높이같은 정보가 포함되지 않으므로 사용자로부터 직접 입력받아야 한다.
- 그러므로 대화상자를 만들어 필요한정보를 수집해야 된다.

파일 불러오기 - 다이얼로그 생성

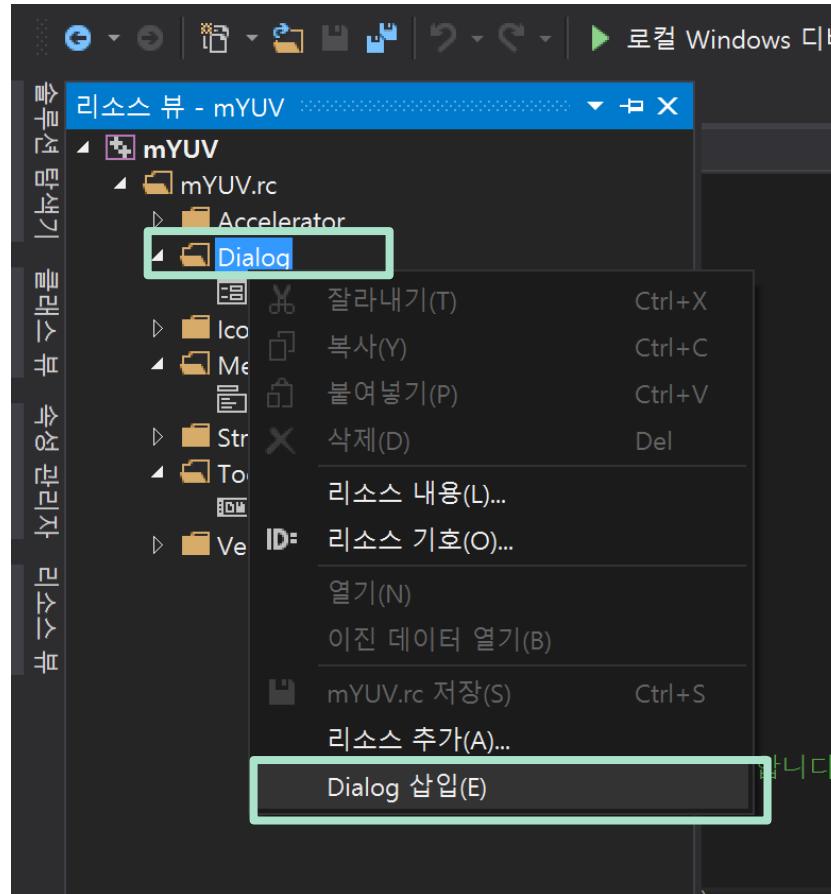
□ 대화 상자 생성 과정

- 솔루션 탐색기 → 리소스파일 → ~.rc를 더블클릭한다.



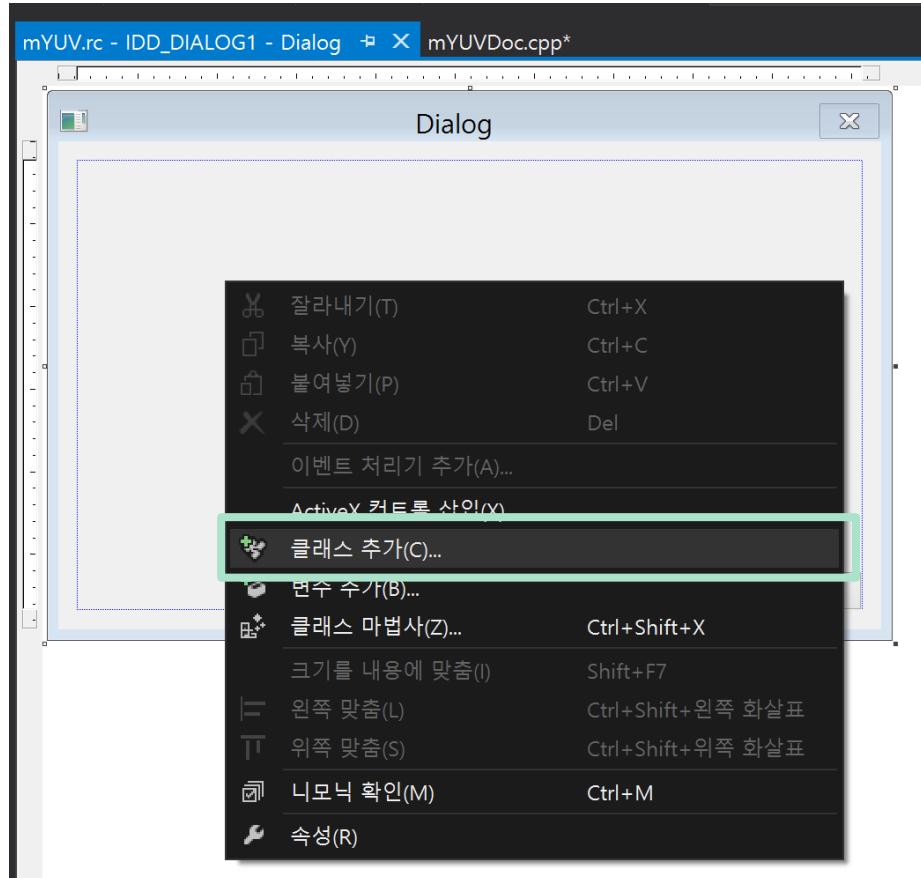
파일 불러오기 - 다이얼로그 생성

- 리소스 뷰에서 Dialog를 선택후 오른쪽 마우스 클릭하여
- Dialog 삽입을 선택한다.



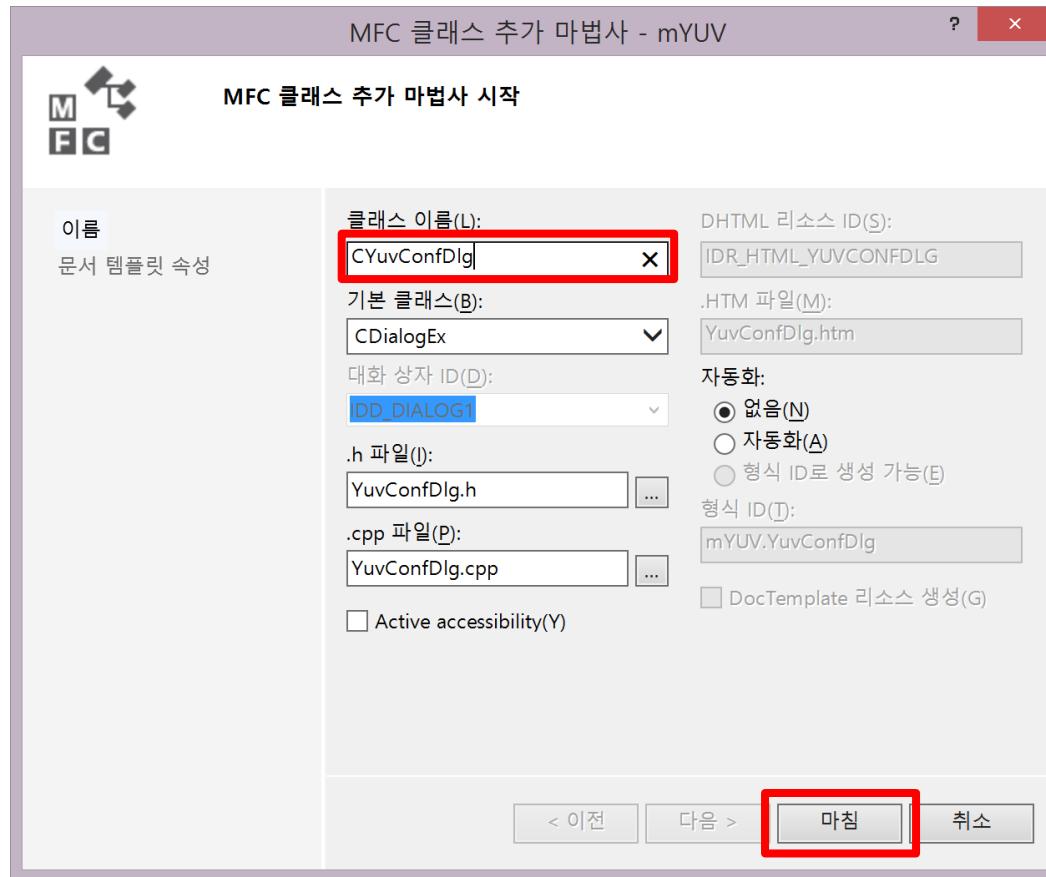
파일 불러오기 - 다이얼로그 생성

- 그러면 아래와 같은 Dialog창이 생성된다.
- Dialog 창에서 오른쪽 마우스 버튼을 누르고 클래스 추가를 누른다.



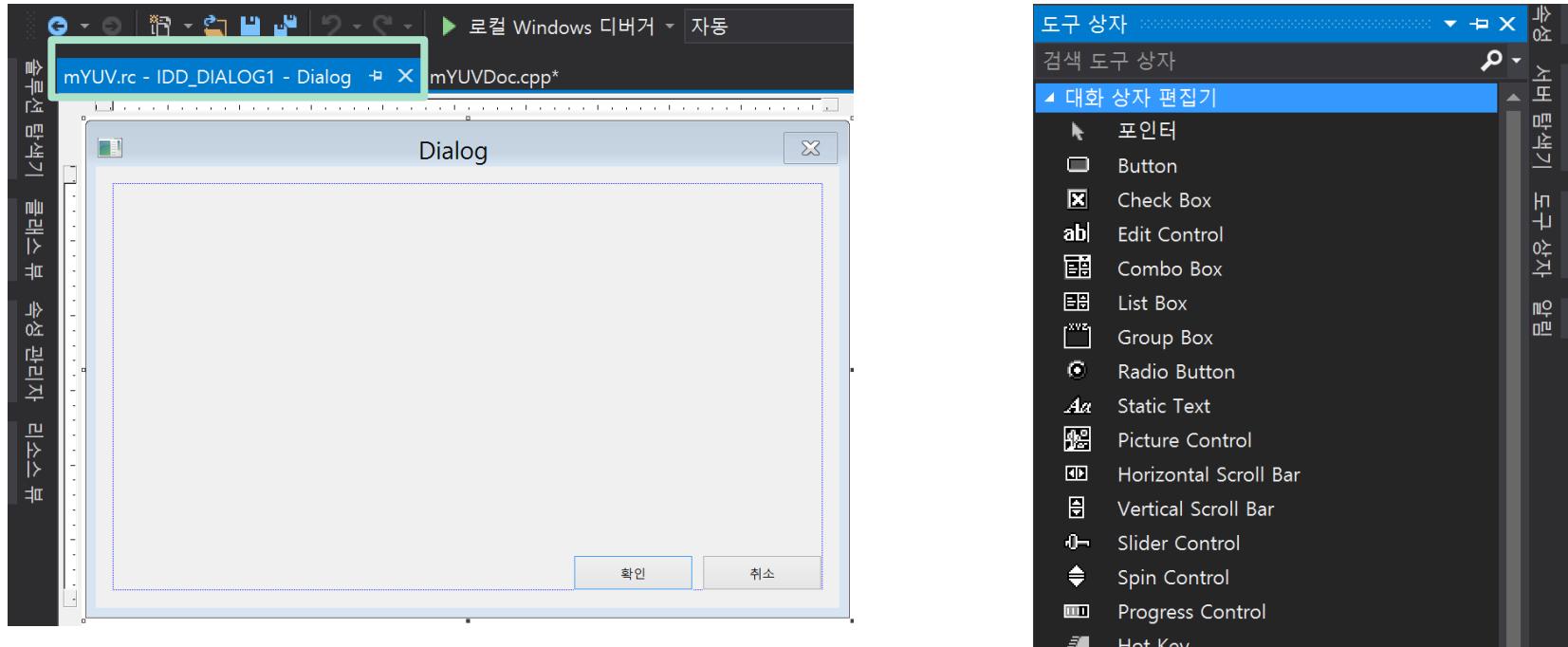
파일 불러오기 - 다이얼로그 생성

- 클래스 이름을 CYuvConfDlg 로 설정하고 마침을 누른다.



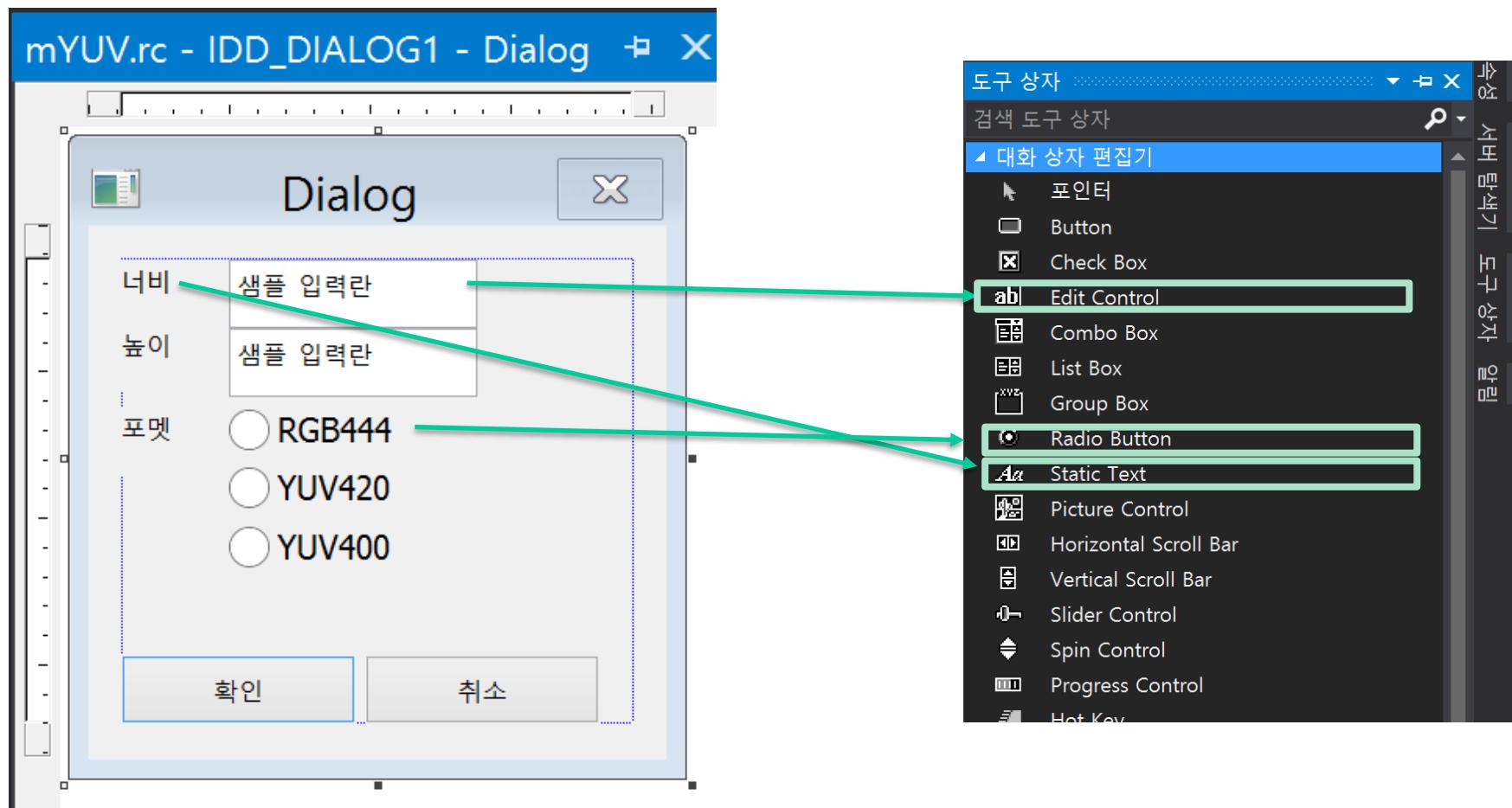
파일 불러오기 - 다이얼로그 생성

- 다시 템에서 ~.rc -IDD_DIALOG1 을 선택한다.
- Ctrl + Alt + X 를 눌러 도구상자를 띄운다.



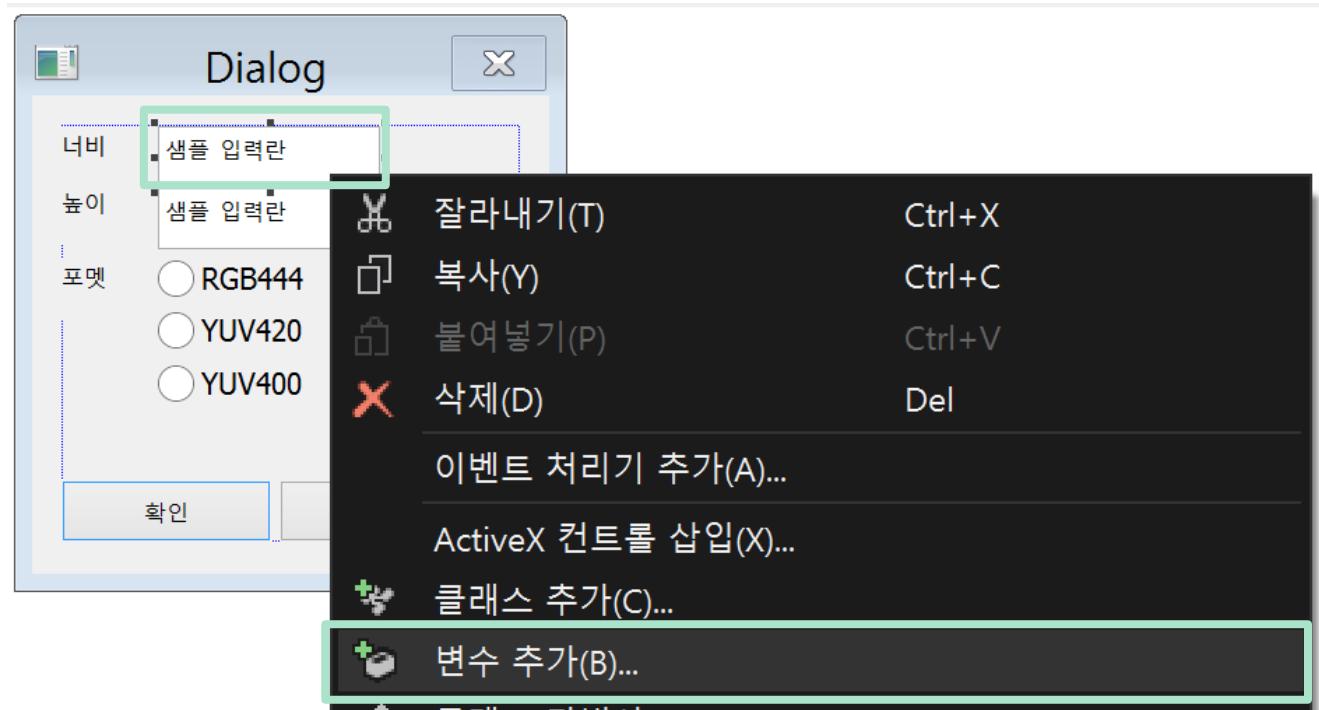
파일 불러오기 - 다이얼로그 생성

▶ 아래와 같이 도구상자를 이용하여 디자인한다.



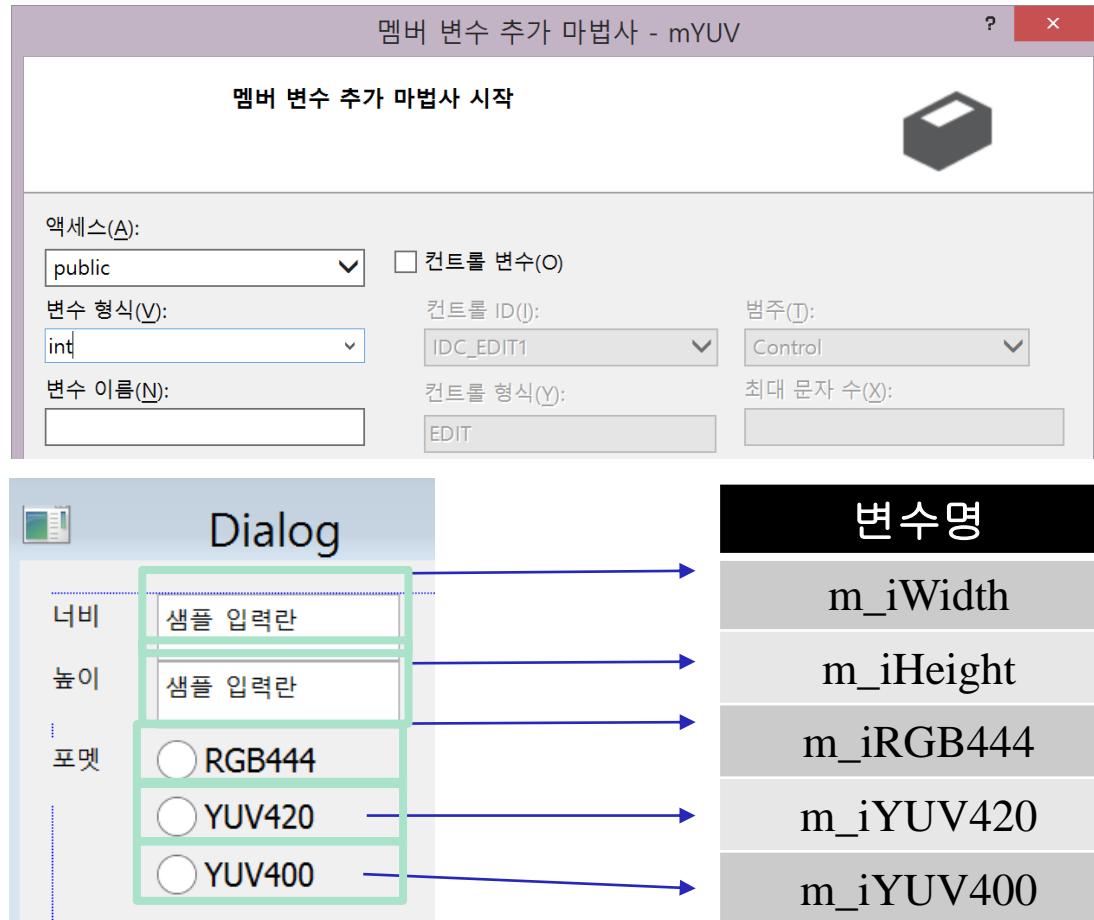
파일 불러오기 - 다이얼로그 생성

- 그리고 2개의 EditControl과 3개의 RadioControl의 변수를 추가한다.



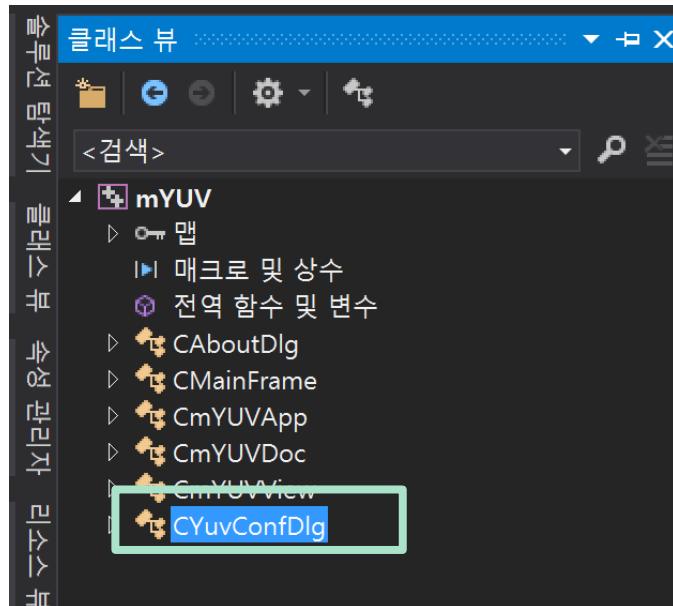
파일 불러오기 - 다이얼로그 생성

- ▶ 변수 추가시 항상 **컨트롤 변수**를 해제하고. 변수형식은 int로 한다.



파일 불러오기 - 다이얼로그 생성

- 클래스 뷰에서 CYuvConfDlg 를 더블클릭하면 소스코드가 popup.
- DoDataExchange 함수를 선택하고 F12를 눌러 함수에 들어간다.



```
class CYuvConfDlg : public CDialogEx
{
    DECLARE_DYNAMIC(CYuvConfDlg)

public:
    CYuvConfDlg(CWnd* pParent = NULL); // 표준 생성
    virtual ~CYuvConfDlg();

// 대화 상자 데이터입니다.
enum { IDD = IDD_DIALOG1 };

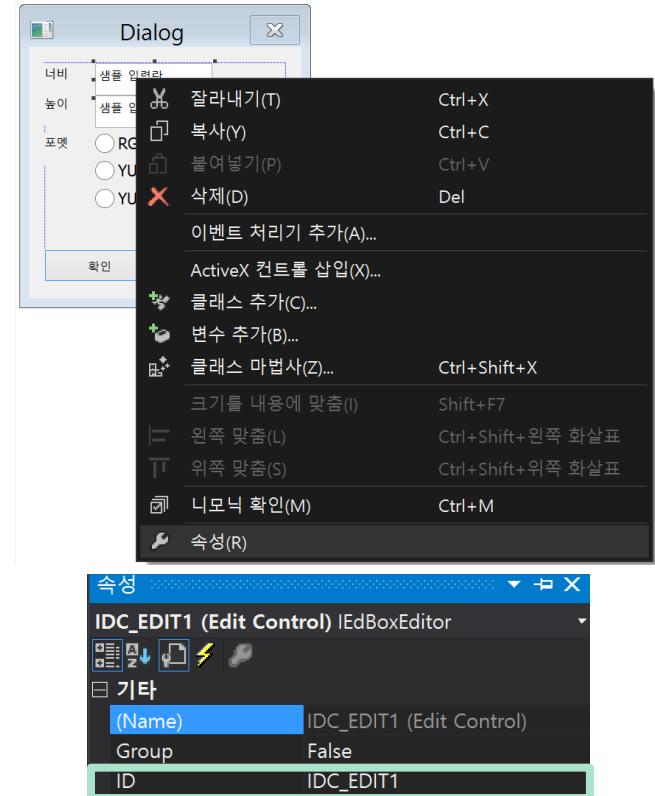
protected:
    virtual void DoDataExchange(CDataExchange* pDX);

    DECLARE_MESSAGE_MAP()
public:
    int m_iRGB444;
    int m_iYUV420;
    int m_iWidth;
    int m_iHeight;
    int m_iYUV400;
};
```

파일 불러오기 - 다이얼로그 생성

- 아래와 같이 코드를 추가합니다.
- IDC_~ 는 컨트롤들의 ID입니다. ID 는 컨트롤을 선택하고 속성에서 확인할 수 있습니다.

```
void CYuvConfDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_EDIT1, m_iWidth);
    DDX_Text(pDX, IDC_EDIT2, m_iHeight);
    DDX_Check(pDX, IDC_RADIO1, m_iRGB444);
    DDX_Check(pDX, IDC_RADIO2, m_iYUV420);
    DDX_Check(pDX, IDC_RADIO3, m_iYUV400);
}
```



파일 불러오기 - 다이얼로그 생성(끝)

- 완성된 다이얼로그를 C~Doc 클래스의 OnFileOpen 함수에 추가하여 아래와 같이 코드를 완성합니다.

```
#include "YuvConfDlg.h"
void CMYUVDoc::OnFileOpen()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    INT_PTR iDialogResult;
    CFileDialog fileDialog(true);

    iDialogResult = fileDialog.DoModal();

    if (iDialogResult != IDOK)
        return;

    CYuvConfDlg yuvConfDialog;
    iDialogResult = yuvConfDialog.DoModal();

    if (iDialogResult != IDOK)
        return;
```

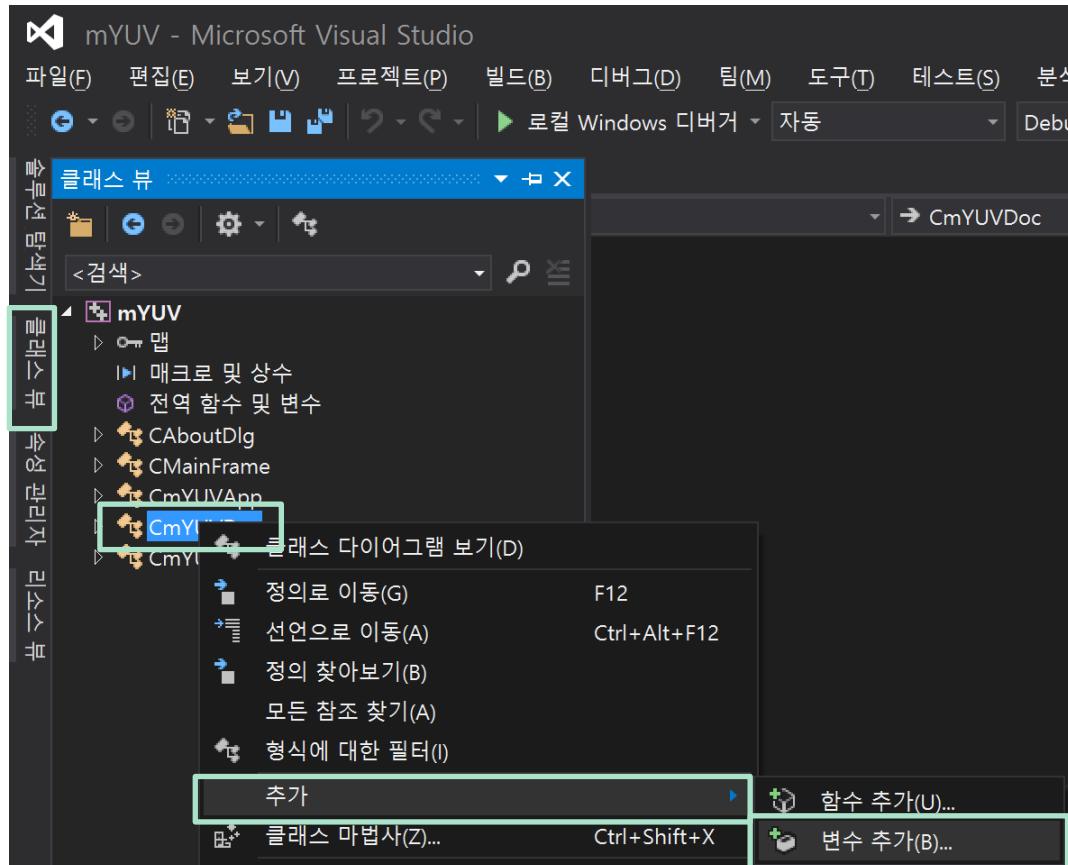
```
enum ImageFormat
{
    NONE = 0,
    RGB444 = 1,
    YUV420 = 2,
    YUV400 = 3,
};

int iWidth = yuvConfDialog.m_iWidth;
int iHeight = yuvConfDialog.m_iHeight;
ImageFormat fmt = NONE;

if (yuvConfDialog.m_iRGB444) fmt = RGB444;
else if (yuvConfDialog.m_iYUV420) fmt = YUV420;
else if (yuvConfDialog.m_iYUV400) fmt = YUV400;
```

BMP 변환

- 이미지 데이터 저장을 위해 맴버변수를 추가한다.
- 클래스뷰 → C~Doc 선택 → 추가 → 변수 추가



BMP 변환

- 바로전의 작업을 아래 표를 참조하여 실행하여 3개의 맴버 변수를 추가한다.

변수 형식	변수 이름
FILE*	m_fdInputImage
BYTE*	m_paBitmapData
BITMAPINFOHEADER	m_bitmapHeader

BMP 변환

- C~Doc 클래스의 OnFileOpen 함수에 코드를 추가합니다.

```
m_fdlInputImage = _tfopen((LPCTSTR)fileDialog.GetPathName(), _T("rb"));

if (m_paBitmapData)
    delete[] m_paBitmapData;

const int iBufferSize = iWidth * iHeight * 3;
m_paBitmapData = new BYTE[iBufferSize];

if (fmt == RGB444)
{
    BYTE b[3];
    BYTE* pDst = m_paBitmapData + iBufferSize - (iWidth * 3);

    for (int i = 0; i < iHeight; i++)
    {
        for (int j = 0; j < iWidth; j++)
        {
            fread(b, sizeof(BYTE), 3, m_fdlInputImage);
            pDst[j * 3 + 0] = b[2];
            pDst[j * 3 + 1] = b[1];
            pDst[j * 3 + 2] = b[0];
        }
        pDst -= (iWidth * 3);
    }
}
```

BMP 변환

```
else if (fmt == YUV420)
{
    BYTE* paYUV = new BYTE[iWidth * iHeight * 3 / 2];
    int cbOffset = iWidth * iHeight;
    int crOffset = cbOffset + (iWidth * iHeight / 4);
    int iWidth2 = iWidth / 2;

    fread(paYUV, sizeof(BYTE), iWidth * iHeight * 3 / 2, m_fdlInputImage);

    int y, cb, cr, r, g, b;
    BYTE* pDst = m_paBitmapData + iBufferSize - (iWidth * 3);

    for (int iY = 0; iY < iHeight; iY++)
    {
        for (int iX = 0; iX < iWidth; iX++)
        {
            int hx = iX / 2;
            int hy = iY / 2;

            y = paYUV[iY * iWidth + iX];
            cb = paYUV[hy * iWidth2 + hx + cbOffset];
            cr = paYUV[hy * iWidth2 + hx + crOffset];

            r = (int)(y + (1.4065f * (cr - 128.f)));
            g = (int)(y - (0.3455f * (cb - 128.f)) - (0.7169f * (cr - 128.f)));
            b = (int)(y + (1.7790f * (cb - 128.f)));

            r = r < 0 ? 0 : (r > 255 ? 255 : r);
            g = g < 0 ? 0 : (g > 255 ? 255 : g);
            b = b < 0 ? 0 : (b > 255 ? 255 : b);

            pDst[iX * 3 + 0] = b;
            pDst[iX * 3 + 1] = g;
            pDst[iX * 3 + 2] = r;
        }
        pDst -= (iWidth * 3);
    }

    delete[] paYUV;
}
```

BMP 변환 (끝)

```
else if (fmt == YUV400)
{
    BYTE b;
    BYTE* pDst = m_paBitmapData + iBufferSize - (iWidth * 3);
    for (int i = 0; i < iHeight; i++)
    {
        for (int j = 0; j < iWidth; j++)
        {
            fread(&b, sizeof(BYTE), 1, m_fdInputImage);

            pDst[j * 3 + 0] = b;
            pDst[j * 3 + 1] = b;
            pDst[j * 3 + 2] = b;
        }
        pDst -= (iWidth * 3);
    }
}
```

$$\begin{aligned} \text{Gray value } k \\ = (r_k, g_k, b_k) \end{aligned}$$

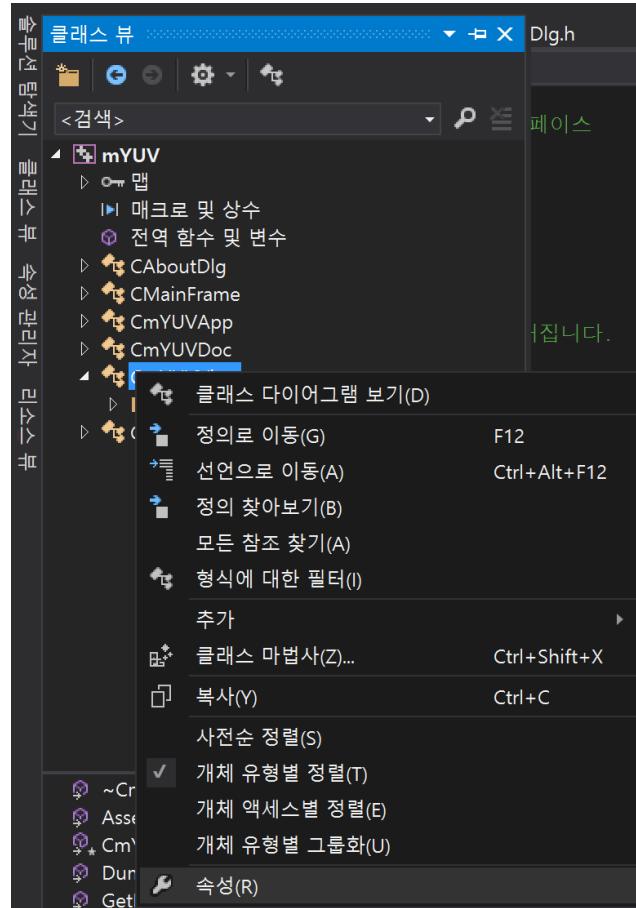
```
m_bitmapHeader.biSize = sizeof(BITMAPINFOHEADER);
m_bitmapHeader.biPlanes = 1;
m_bitmapHeader.biBitCount = 24;
m_bitmapHeader.biCompression = BI_RGB;
m_bitmapHeader.biSizeImage = 0;
m_bitmapHeader.biXPelsPerMeter = 0;
m_bitmapHeader.biYPelsPerMeter = 0;
m_bitmapHeader.biClrUsed = 0;
m_bitmapHeader.biClrImportant = 0;
m_bitmapHeader.biWidth = iWidth;
m_bitmapHeader.biHeight = iHeight;

fclose(m_fdInputImage);

UpdateAllViews(0);
```

출력

- BMP로 변환 저장된 데이터를 출력하기 위해선 C~View 클래스의 OnDraw함수를 수정해야한다.



```
mYUVView.h  mYUVDoc.h  mYUVDoc.cpp  YuvConfDlg.h  YuvConfDlg.cpp
mYUVView.h
// mYUVView.h : CmYUVView 클래스의 인터페이스
//

#pragma once

class CmYUVView : public CView
{
protected: // serialization에서만 만들어집니다.
    CmYUVView();
    DECLARE_DYNCREATE(CmYUVView)

    // 특성입니다.
public:
    CmYUVDoc* GetDocument() const;

    // 작업입니다.
public:

    // 재정의입니다.
public:
    virtual void OnDraw(CDC* pDC); // 이 뷰를 그리기 위해 재정의되었습니다.
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
```

출력

- C~View :: OnDraw 함수를 수정한다.

```
void CmYUVView::OnDraw(CDC* pDC)
{
    CmYUVDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    // TODO: 여기에 원시 데이터에 대한 그리기 코드를 추가합니다.
    if (pDoc->m_paBitmapData)
    {
        int iWidth = pDoc->m_bitmapHeader.biWidth;
        int iHeight = pDoc->m_bitmapHeader.biHeight;
        |
        ::StretchDIBits(pDC->GetSafeHdc(), 0, 0, iWidth, iHeight, 0, 0,
                        iWidth, iHeight, pDoc->m_paBitmapData,
                        (LPBITMAPINFO)&pDoc->m_bitmapHeader, DIB_RGB_COLORS, SRCCOPY);
    }
}
```

결과

