

HTC-3500 Series Vision Camera

Multi Vision Camera Development Manual

Ver 2.1

2013. 06. 25



<http://www.hontek.co.kr>

<http://cafe.naver.com/hontek>

목 차

1. 개요	4
2. Multi Vision Camera Specification	5
2.1 General feature	5
2.2 PC Requirements	5
2.3 HTC-3500CB - Contents	6
2.4 HTC-3500CB - Dimension and Description	6
2.5 HTC-3500CB - Camera Interface	7
2.6 HTC-3500CB - Trigger / Strobe Connector Port	7
2.7 HTC-3500CB - Electrical Operating Condition	8
2.8 HTC-3500R - Contents	9
2.9 HTC-3500R - Camera Interface	9
2.10 HTC-3500R - Dimension and Description	10
3. 개발환경	11
3.1 개발 요구사항	11
3.2 Development Architecture	11
4. Software Architecture	12
4.1 장치 관리자	12
4.2 DirectShow Filter 설치	13
4.3 프로그램 실행	14
4.4 프로그램 종료	15

목 차

4.5 Play / Stop	16
4.6 Resolution Setting	17
4.7 Fitting Window	19
4.8 Image Signal Processor	20
4.9 Snapshot	22
4.10 OpenCV	23
4.11 Program Info	25
5. API 설명	26
5.1 HVR_Set_ISP	26
5.2 HVR_Set_ISP2	27
5.3 BmpSmoothingMask	28
5.4 BmpContrast	29
5.5 BmpThreshold	30
5.6 BmpSharpening	31
5.7 CB_Init	32
6. Sample Source 설명	33
7. 기타 문의 사항	38

1. 개요

본 문서는 멀티비전카메라를 이용하기 위한 사용자의 이해를 돕기 위해 만든 문서로 하드웨어 구성, 소프트웨어 구성, 프로그램 개발에 대한 설명을 나열 하였습니다. 기타 추가적인 질문사항이 있으시면 아래와 같은 방법으로 문의 하시기 바랍니다.

(주) 혼텍

주소 : 경기도 성남시 중원구 상대원동 SK[®]테크노파크 테크동 810호

전화 : 031-776-3897, 3899

팩스 : 031-776-3898

홈페이지 : <http://www.hontek.co.kr>

2. Multi Vision Camera Specification

2.1 General feature

☞ PC Interface	USB 3.0 (5Gbps)
☞ Sensor video data format	8bit Bayer
☞ Sensor command interface	I2C protocol
☞ Compression	압축 안함
☞ Image Processing	Host 프로그램 구현 또는 센서 ISP 컨트롤 가능
☞ Multi-Camera connect	동일 PC 최대 4대 연결 가능

2.2 PC Requirements

☞ CPU	Pentium4 2.8GHz이상 (3.0GHz 이상 추천)
☞ RAM	2GB 이상
☞ USB 3.0 Host Controller	Intel Controller chip 권장
☞ Operating System	Windows 2000 SP4 이상 / WindowsXP SP2이상 Windows Vista / Windows 7

2. Multi Vision Camera Specification

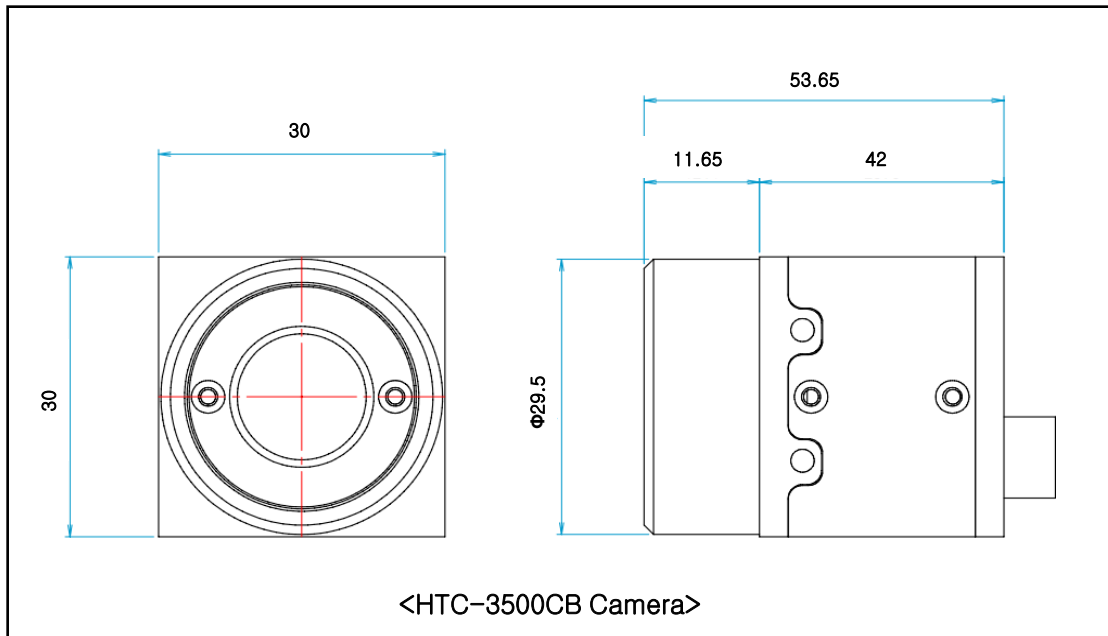
2.3 HTC-3500CB - Contents

- ☞ HTC-3500CB 3.0 Camera
- ☞ Micro USB 3.0 Cable (3M)
- ☞ User's manual / Program CD
- ☞ Trig/Strobe 원형 Cable (4Pin)



2.4 HTC-3500CB - Dimension and Description

- ☞ Camera Body Size : 30(w) x 30(h) x 53(D) mm

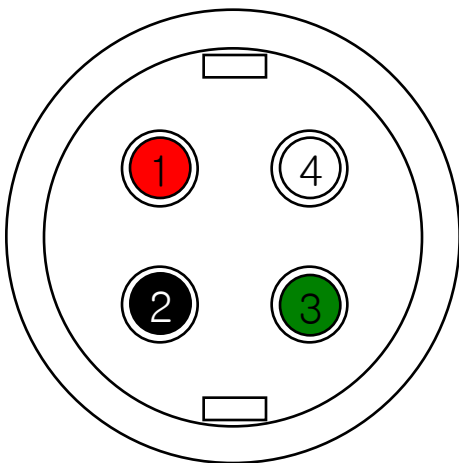


2. Multi Vision Camera Specification

2.5 HTC-3500CB - Camera Interface



2.6 HTC-3500CB - Trigger / Strobe Connector Port



<Camera Side>

Pin	Signal Name
1	Trigger +
2	Trigger -
3	Strobe +
4	Strobe -

* Wire Color

1 : Red

2 : Black

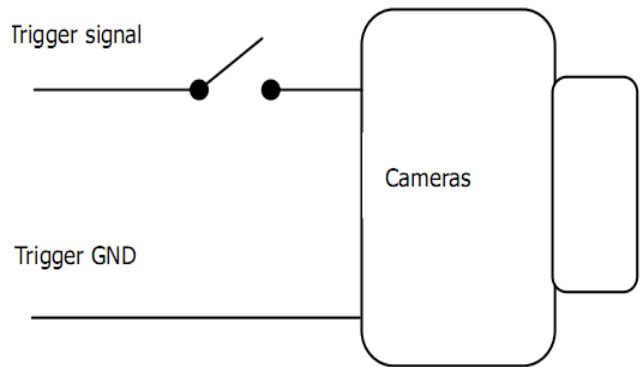
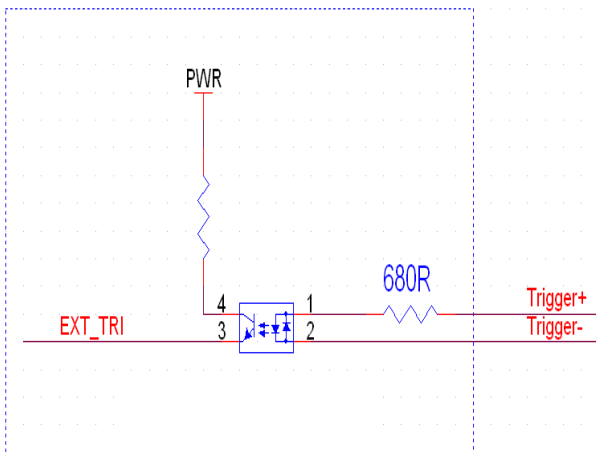
3 : Green

4 : White

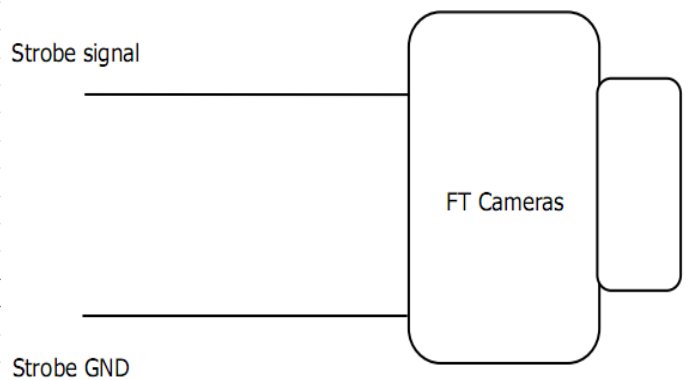
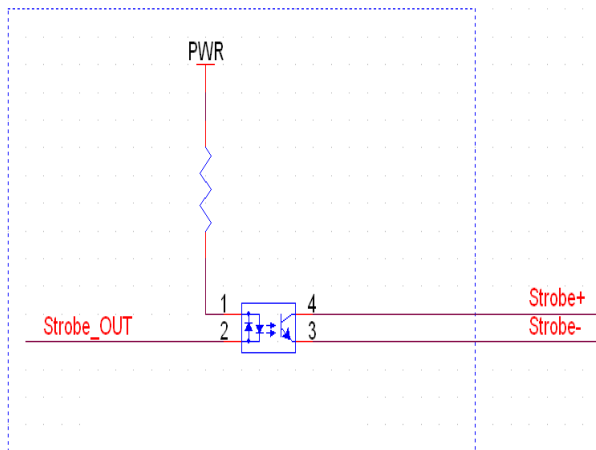
2. Multi Vision Camera Specification

2.7 HTC-3500CB - Electrical Operating Condition

<Trigger>



<Strobe>



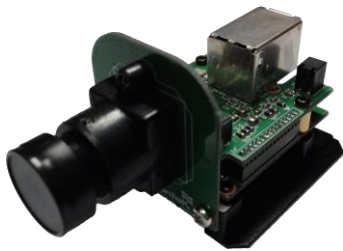
2. Multi Vision Camera Specification

2.8 HTC-3500R - Contents

☞ HTC-3500R 3.0 Camera

☞ USB 3.0 A/B Cable (3M)

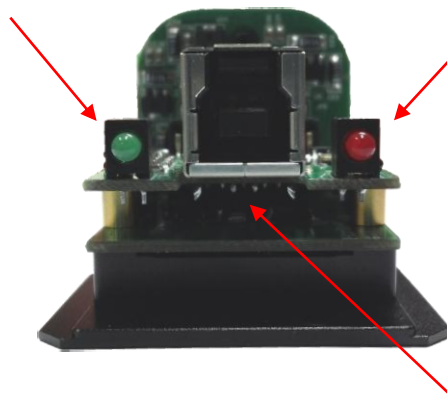
☞ User's manual / Program CD



2.9 HTC-3500R - Camera Interface

동작 LED

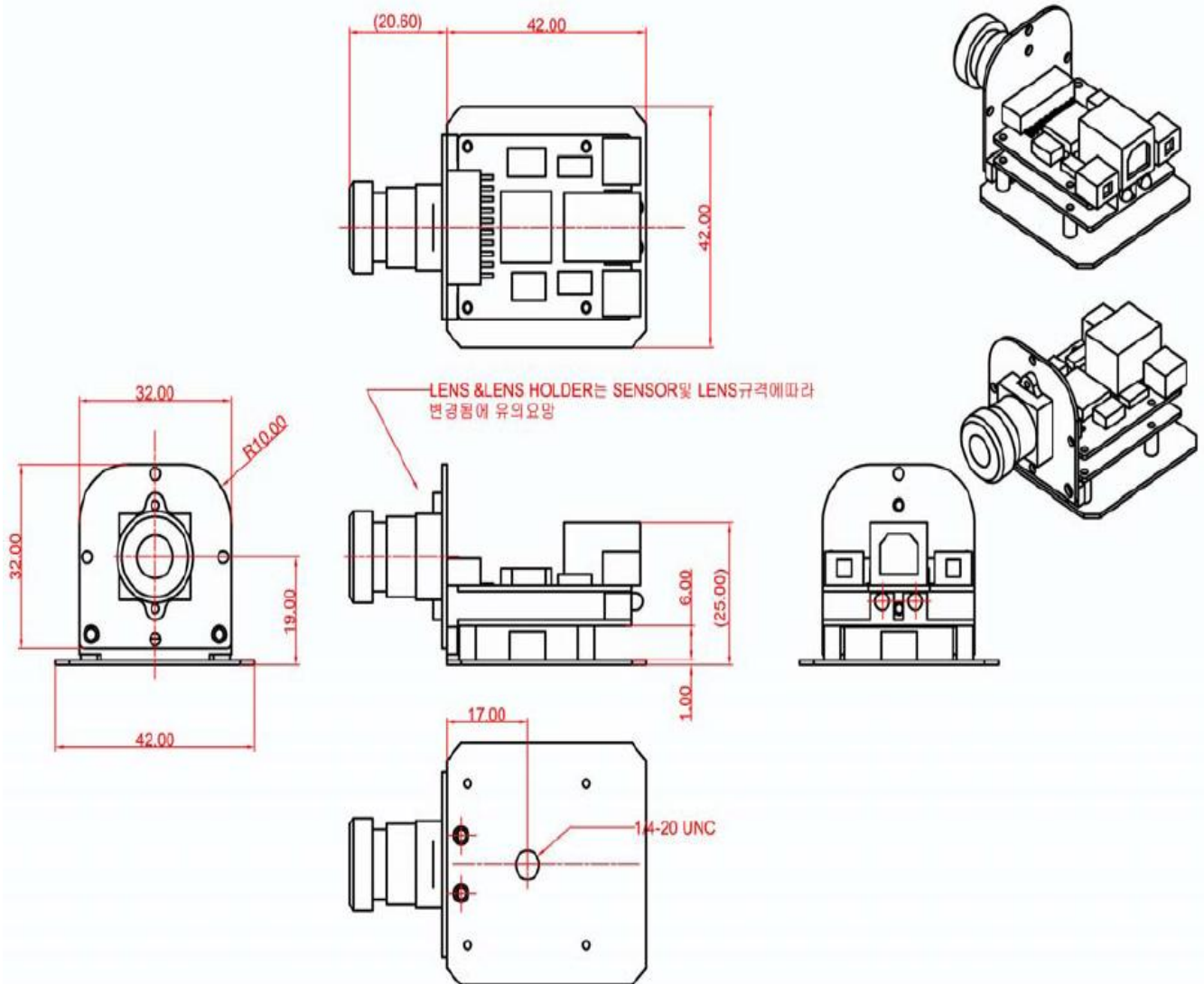
POWER LED



USB 3.0 Connector

2. Multi Vision Camera Specification

2.10 HTC-3500R - Dimension and Description



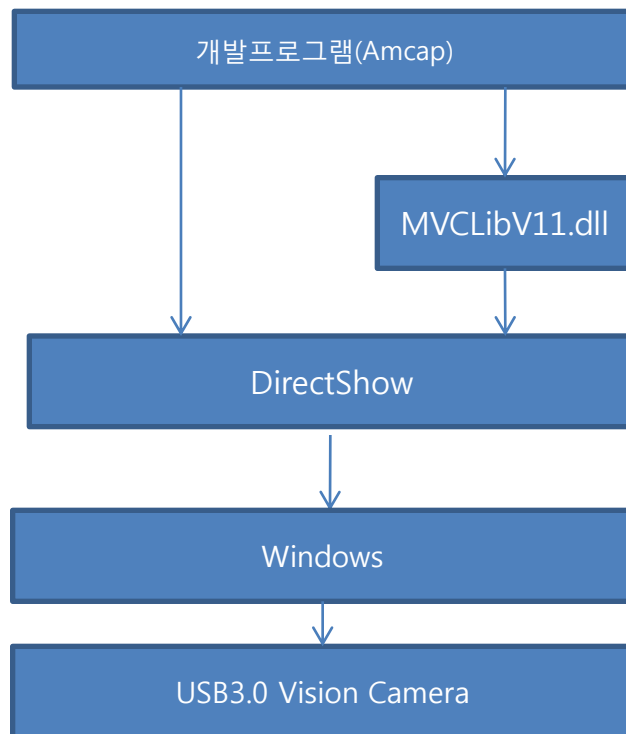
<HTC-3500R Camera>

3. 개발 환경

3.1 개발 요구사항

- ☞ 운영체제 : Microsoft Windows 2000 SP4 이상, WindowsXP SP2 이상,
Windows Vista, Windows 7
- ☞ 컴파일러 : Visual C++ 6.0
- ☞ 기타 : DirectX SDK

3.2 Development Architecture



4. Software Architecture

4.1 장치 관리자

카메라를 연결하면 “장치 관리자”=>”이미지장치”에 다음과 같이 나타납니다.(OS에 따라 다른 이름으로 나타날 수 있습니다)

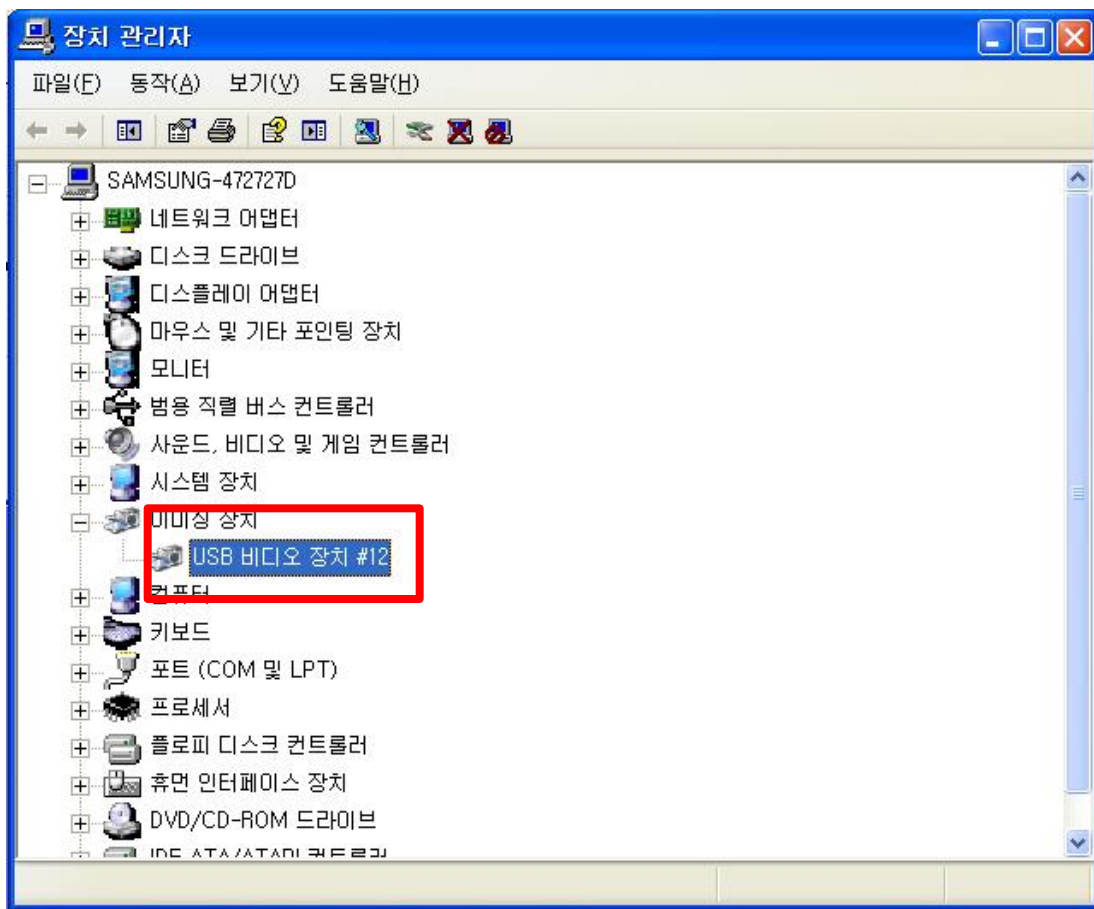


Fig 1. 장치 관리자

4. Software Architecture

4.2 DirectShow Filter 설치

멀티 비전 카메라를 사용하기 위해서는 DirectShow Filter를 설치해야 합니다.

FilterSetup.exe를 실행하고 Register버튼을 클릭하시면 자동으로 필터가 설치됩니다.



Fig 2. FilterSetup.exe



Fig 3. 필터등록

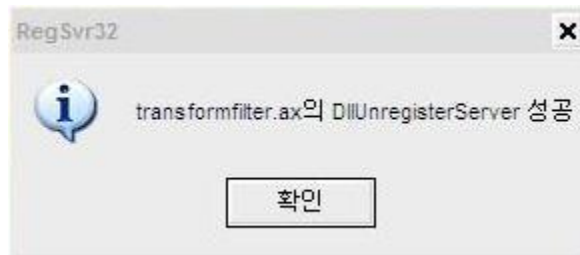


Fig 4. 필터등록해제

4. Software Architecture

4.3 프로그램 실행

멀티 비전 프로그램을 실행시키면 다음과 같은 화면이 나타난다.

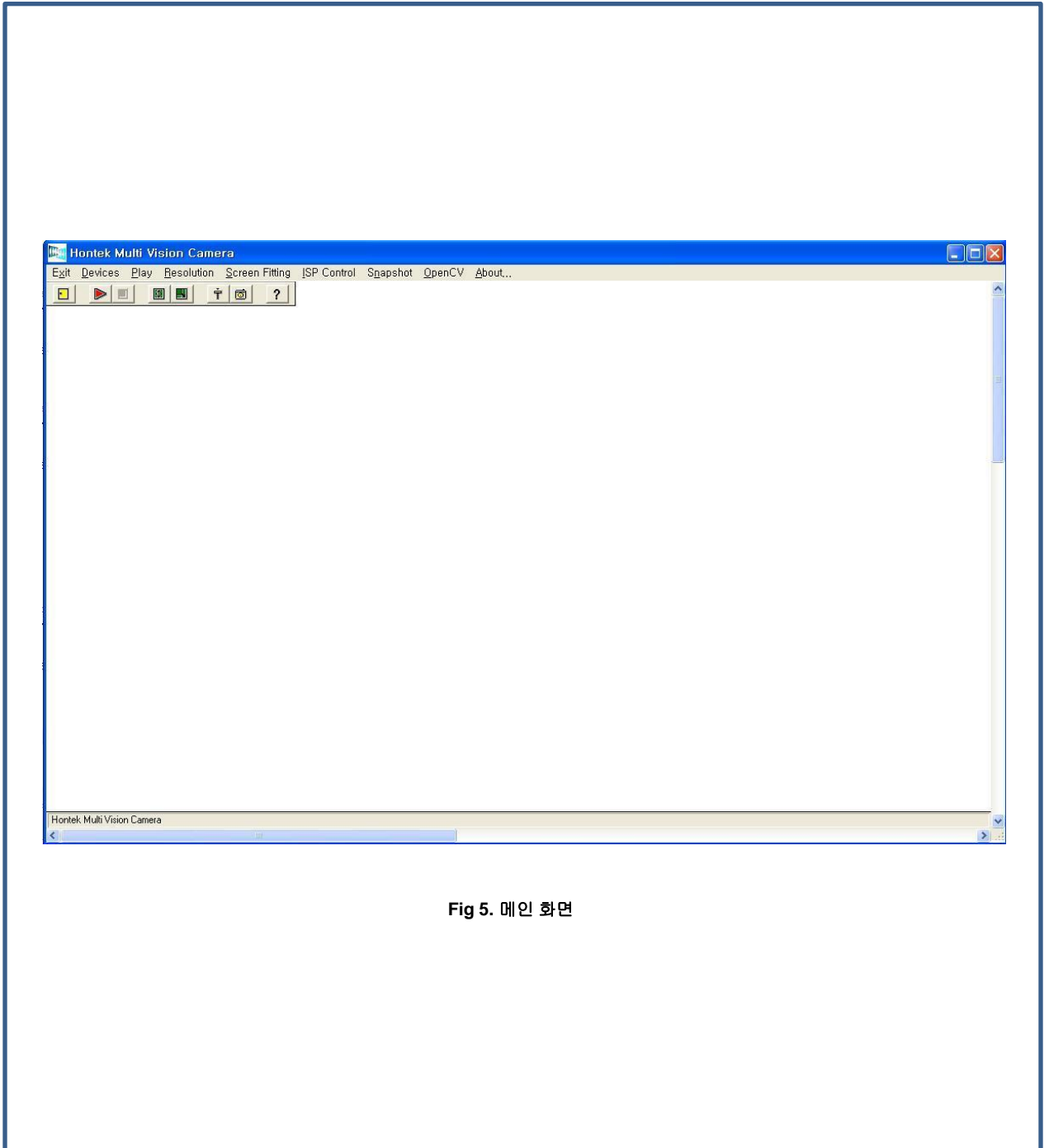


Fig 5. 메인 화면

4. Software Architecture

4.4 프로그램 종료

사용 프로그램을 종료합니다.

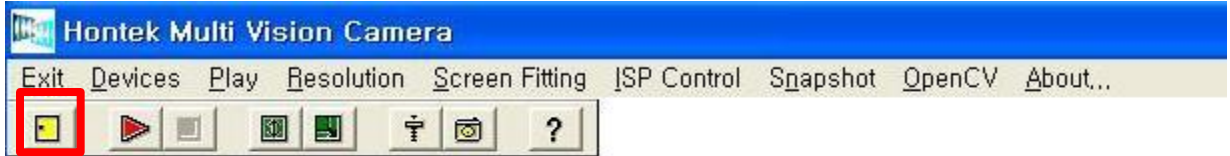


Fig 6. 프로그램 종료

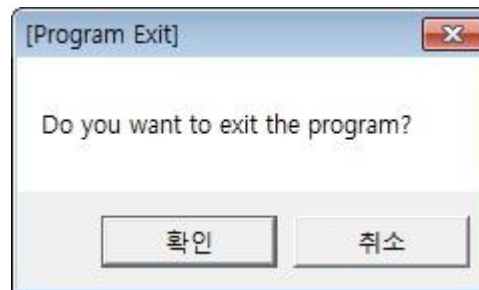


Fig 7. 종료 메시지

Exit 버튼을 선택하거나, Exit메뉴를 선택하면 프로그램 종료 확인 창이 나타나며, 이때 확인버튼을 선택하면 프로그램이 종료됩니다.

4. Software Architecture

4.5 Play / Stop

영상을 Play 하거나 Stop 할 수 있습니다.



Fig 8. Play

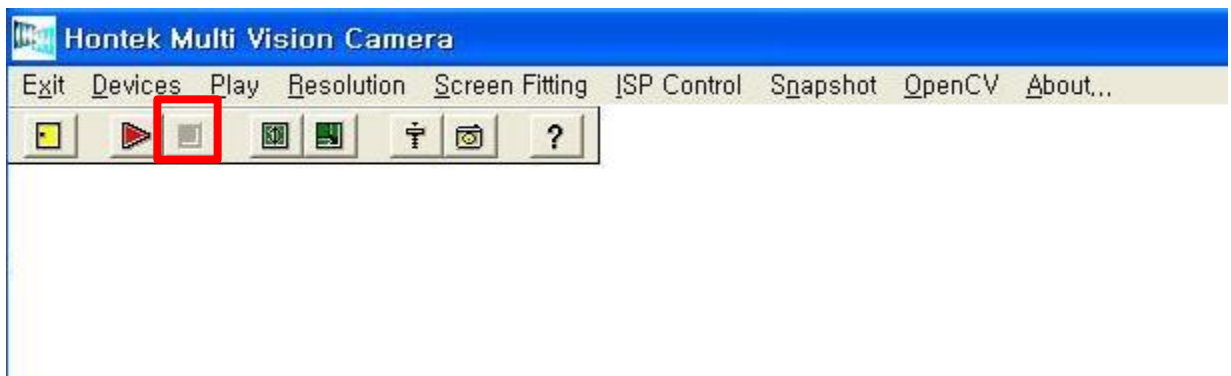


Fig 9. Stop

Play : 카메라를 동작시킨다.

Stop : 카메라의 동작을 멈춘다.

4. Software Architecture

4.6 Resolution Setting

해상도를 변경하거나 Vblank, Hblank를 변경한다.



Fig 10. Resolution

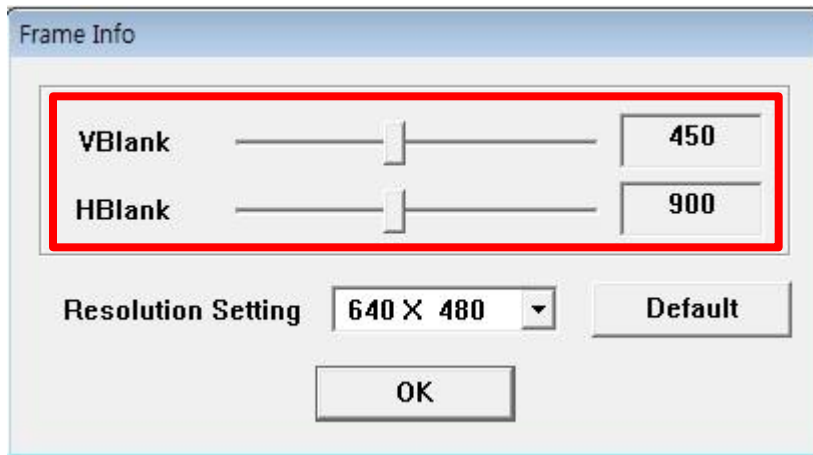


Fig 11. Vblank, HBlank

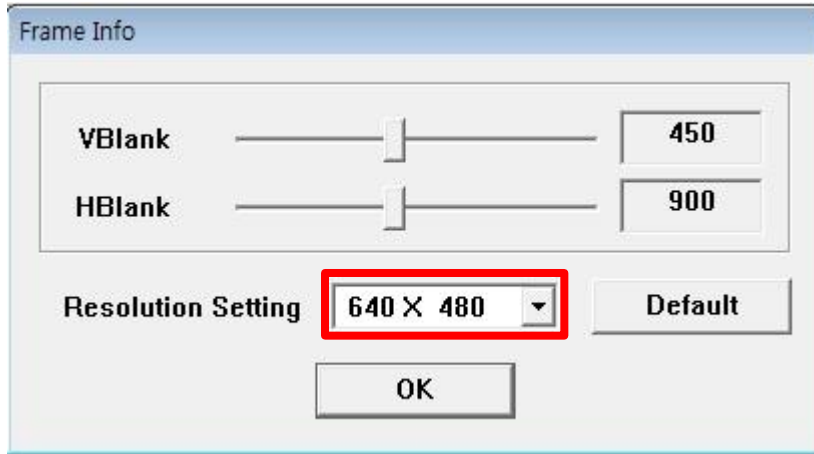
Vblank, Hblank : 영상 프레임 사이에 간격을 주어 컴퓨터에서 영상을 처리할 수 있는 시간을 준다.

Default Button : Vblank와 Hblank를 기본값으로 설정한다.

4. Software Architecture

4.6 Resolution Setting

해상도를 변경하거나 Vblank, Hblank를 변경한다



해상도	Width X Height
QSXGA	2592 X 1944
QXGA	2048 X 1536
SXGA	1280 X 1024
XGA	1024 X 768
SVGA	800 X 600
VGA	640 X 480

Fig 12. Resolution Setting

해상도를 변경하면 센서로부터 나오는 영상의 크기가 변경 됩니다.

4. Software Architecture

4.7 Fitting Window

영상을 화면 크기에 맞춥니다.

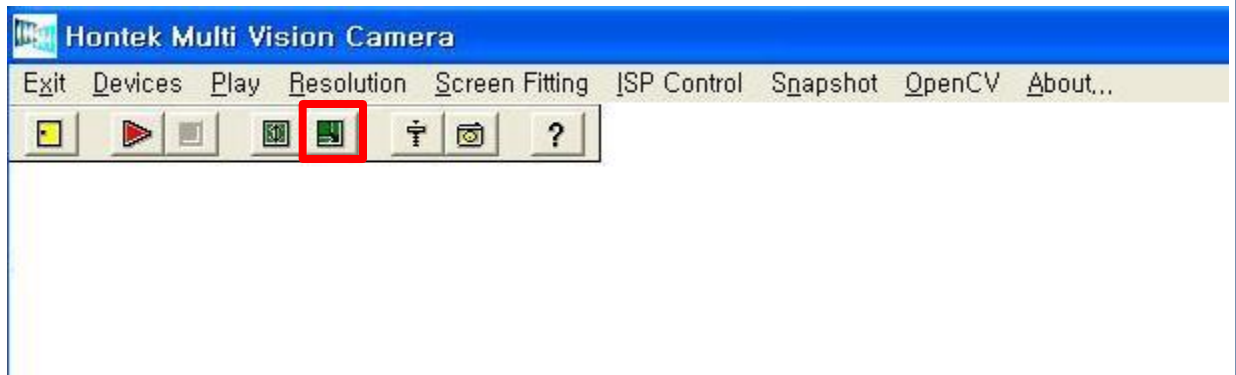


Fig 13. Resolution

Fitting Window를 선택하면 카메라 영상을 화면의 크기에 맞추어
화면에 출력합니다.

4. Software Architecture

4.8 Image Signal Processor

이미지에 효과를 추가한다.

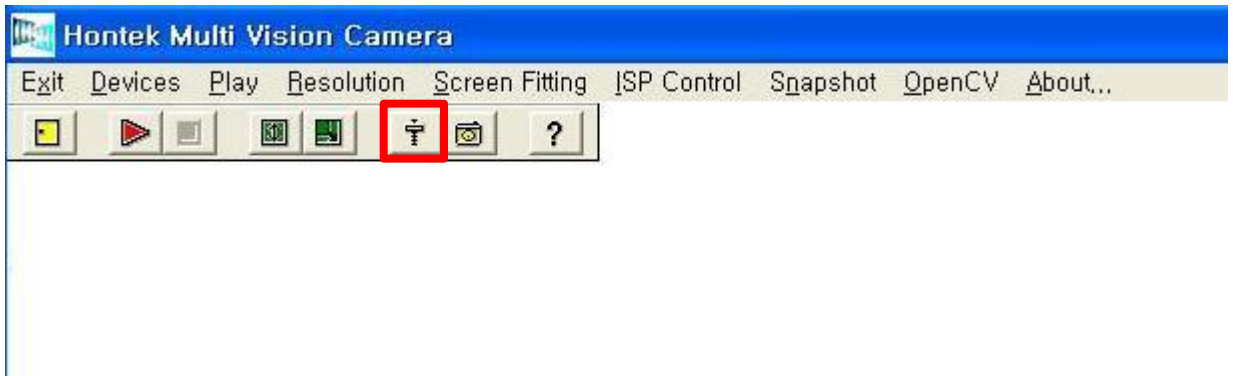


Fig 14. ISP

이미지에 Gain, Brightness, Contrast, Sharpness 등의 효과를 추가한다.

4. Software Architecture

4.8 Image Signal Processor

이미지에 효과를 추가한다

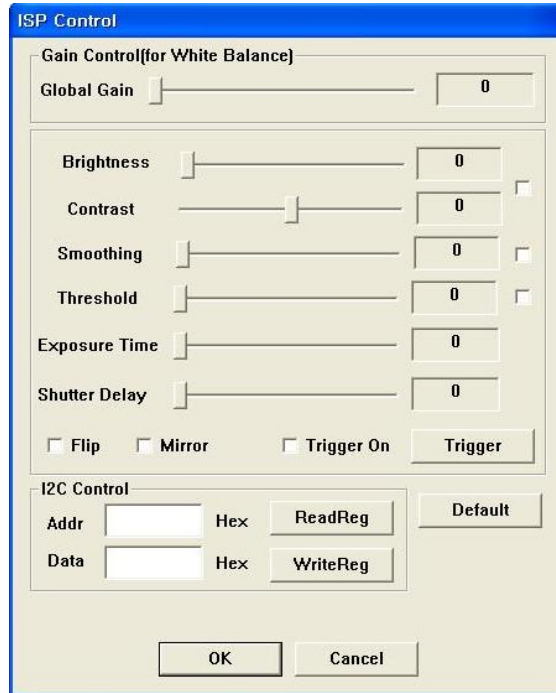


Fig 15. ISP Control

Gain : 카메라로부터 입력되는 영상의 GAIN에 대한 가중치를 설정
Brightness : 밝기 값 설정
Contrast: 대비 값 설정
Smoothing : 화면을 매끄럽게 만든다.
Threshold : 한계점 설정
Exposure Time : 노출시간 설정
Shutter Delay : 이미지 사이의 간격
Flip : 뒤집기 효과
Mirror : 거울 효과
Trigger : 신호가 있어야만 카메라로부터 영상을 받을 수 있다.
Default: 기본 값으로 설정한다.
ReadReg, WriteReg : 센서 세팅 값을 확인, 변경

4. Software Architecture

4.9 Snapshot

정지영상을 저장한다.



Fig 16. Snapshot

정지 영상을 저장한다.

저장 형식 : bmp, jpg

4. Software Architecture

4.10 OpenCV

OpenCV함수를 이용하여 영상을 출력한다..



Fig 17. Snapshot

OpenCV 함수를 이용하여 영상을 처리한다.

4. Software Architecture

4.10 OpenCV

OpenCV 이미지에 효과를 추가한다



Fig 18. ISP Control

Edge-canny : Canny Edge 외곽선 추출

Blur : 영상을 흐릿하게 만듦

4. Software Architecture

4.11 Program Info

프로그램에 대한 정보를 보여준다.

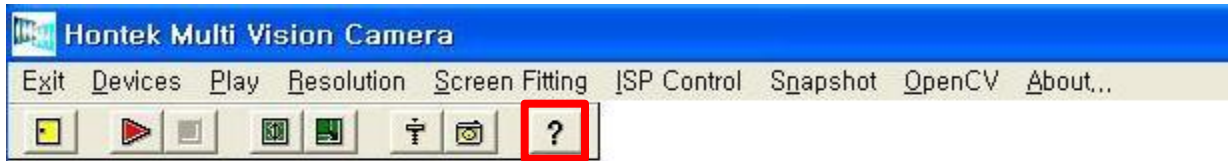


Fig 19. Program Info

5. API 설명

5.1 HVR_Set_ISP

카메라의 설정 값을 변경한다.

구분	내용	
Syntax	HVR_Set_ISP(IAMCameraControl *pCameraControl, int mode, LONG lvalue, LONG *IMin, LONG *IMax, LONG *IStep, LONG *gDrvIndex, LONG *IPropertyRangeFlags);	
Parameter	pCameraControl	카메라 컨트롤 핸들
	mode	4: ExposureTime 5: Shutter Delay 6: Hblank 7: Vblank 8: Trigger 9: 연결 USB Speed
	lvalue	각 모드 값
	IMin	Gain최소값 (mode 9에서만 사용)
	IMax	Gain최대값(mode 9에서만 사용)
	IStep	Gain증분값(mode 9에서만 사용)
	gDrvIndex	USB Speed (gDrvIndex / 0x10) (mode 9에서만 사용)
	IPropertyRangeFlags	제어방식(자동, 수동)
Return Value		
Sample	<pre>gcap.pVCap- >QueryInterface(IID_IAMCameraControl,(void**)&pCameraControl); HVR_Set_ISP(pCameraControl, 6, IHBlank, 0, 0, 0, 0, 0);</pre>	

5. API 설명

5.2 HVR_Set_ISP2

비디오 설정 값을 변경한다.

구분	내용	
Syntax	HVR_Set_ISP2(IAMVideoProcAmp *pCameraControl, int mode, LONG value, LONG lvalue);	
Parameter	pCameraControl	비디오 컨트롤 핸들
	mode	9: Gain
	value	Gain값
	lvalue	0(Reserved)
Return Value		
Sample	gcap.pVCap->QueryInterface(IID_IAMVideoProcAmp, (void**)&pVideoControl); HVR_Set_ISP2(pVideoControl, 9, IAWB_R, 0);	

5. API 설명

5.3 BmpSmoothingMask

영상에 Smooth 효과를 준다.

구분	내용	
Syntax	BmpSmoothingMask(BYTE *pBuf, LONG ImSmoothing);	
Parameter	pBuf	영상데이터가 들어있는 버퍼의 포인터
	ImSmoothing	Matrix값(1 ~ 4)
Return Value		
Sample	BmpSmoothingMask(pRGBBuffer1, ImSmoothing, IMAGE_SIZE_X, IMAGE_SIZE_Y);	

5. API 설명

5.4 BmpContrast

영상에 밝기 값과 대비 값을 변경한다.

구분	내용	
Syntax	BmpContrast(BYTE *CBBuf, BYTE *pBuf, double Brightness, double Contrast);	
Parameter	CBBuf	영상데이터가 들어있는 버퍼의 포인터
	pBuf	효과를 적용한 영상데이터를 저장할 버퍼
	Brightness	밝기값
	Contrast	대비값
Return Value		
Sample	BmpSmoothingMask(pRGBBuffer1, ImSmoothing, IMAGE_SIZE_X, IMAGE_SIZE_Y);	

5. API 설명

5.5 BmpThreshold

영상에 한계점 효과를 준다.

구분	내용	
Syntax	BmpThreshold(BYTE *pBuf, DWORD lmThreshold, unsigned int nWidth, unsigned int nHeight);	
Parameter	pBuf	영상데이터가 들어있는 버퍼의 포인터
	lmThreshold	한계점값
	nWidth	영상의 폭
	nHeight	영상의 너비
Return Value		
Sample	BmpThreshold(pRGBBuffer1, lmThreshold, IMAGESIZE_X, IMAGESIZE_Y);	

5. API 설명

5.6 BmpSharpening

영상의 선명도를 변경한다.

구분	내용	
Syntax	BmpSharpening (BYTE *pBuf, LONG lmSharpening, unsigned int nWidth, unsigned int nHeight);	
Parameter	pBuf	영상데이터가 들어있는 버퍼의 포인터
	lmSharpening	선명도(1 ~ 5)
	nWidth	영상의 너비
	nHeight	영상의 높이
Return Value		
Sample	CB_Init(nlmBrightness, nlmContrast, CBBuf);	

5. API 설명

5.7 CB_Init

영상의 밝기 값과 대비 값 처리를 위해 초기작업을 수행한다.

구분	내용	
Syntax	CB_Init(double Brightness, double Contrast, BYTE *CBBuf);	
Parameter	Brightness	밝기값
	Contrast	대비값
	CBBuf	영상데이터가 들어있는 버퍼의 포인터
Return Value		
Sample	CB_Init(nlmBrightness, nlmContrast, CBBuf);	

6. SAMPLE Source 설명

Amcap 프로그램을 기본으로 제작되었습니다.

Amcap.cpp : DirectShow의 메인 함수가 모인 파일로 필터생성, 프리뷰, 각 메뉴 처리 등, 주요 기능을 담당하고 있는 파일입니다.

1. **INIReadStringVision** : ini 파일에서 필요한 정보를 읽어오는 함수

LPCTSTR lpAppName : 값을 가져올 키가 속해있는 [섹션]의 문자열

LPCTSTR lpKey : 값을 가져올 키

LPCSTR lpDefault : 해당되는 값이 없을 경우, 기본값으로 리턴할 문자열

LPTSTR lpReturn : 문자열을 리턴할 String

DWORD nSize : 가져올 키값의 길이

2. **INIWriteStringVision** : ini 파일에 필요한 정보를 저장하는 함수

LPCTSTR lpAppName : 값을 저장할 키가 속해있는 [섹션]의 문자열

LPCTSTR lpKey : 값을 저장할 키

LPCSTR str : 저장할 문자열

3. **AppInit** : 프로그램 시작 시 실행되는 함수

HINSTANCE hInst : 현재의 인스턴드 핸들값

HINSTANCE hPrev : 이전의 인스턴드 핸들값

int sw : 윈도우를 화면에 출력하는 방법

6. SAMPLE Source 설명

4. AppWndProc : 윈도우 메시지를 처리하는 함수

HWND hwnd : 윈도우 핸들

UINT msg: 메시지 넘버

WPARAM wParam : message-dependent

LPARAM lParam : message-dependent

5. ResizeWindow : 윈도우의 크기를 변경하는 함수

int w: 윈도우의 폭

int h: 윈도우의 높이

6. BuildPreviewGraph : Preview graph 를 build하는 함수

7. StartPreview : 미리보기 기능 수행함수

8. StopPreview : 미리보기 기능 중지함수

9. ChooseDevices : 선택된 카메라를 초기화 하는 함수

IMoniker *pmVideo : 비디오 모티커 인터페이스

IMoniker *pmAudio: 오디오 모티커 인터페이스

10. AppCommand : 메뉴 처리 함수

HWND hwnd : 윈도우 핸들

UINT msg: 메시지 넘버

WPARAM wParam : message-dependent

LPARAM lParam : message-dependent

6. SAMPLE Source 설명

11. **ISP_Function** : ISP 처리 함수

HWND hwnd : 윈도우 핸들

12. **Resolution** : 해상도 변경함수

HWND hwnd : 윈도우 핸들

13. **ISPProc** : ISP Control 다이알로그 처리함수

HWND hwnd : 윈도우 핸들

UINT msg: 메시지 넘버

WPARAM wParam : message-dependent

LPARAM lParam : message-dependent

14. **FrameProc**: Vblank, Hblank 해상도를 설정하는 다이알로그 처리 함수

HWND hwnd : 윈도우 핸들

UINT msg: 메시지 넘버

WPARAM wParam : message-dependent

LPARAM lParam : message-dependent

15. **Read_User_Control_Set** : 미지 저장한 Control 값들을 적용

6. SAMPLE Source 설명

SampleGrabberCB.cpp : 카메라에서 프로그램으로 영상데이터가 전달되면 이벤트가 발생하며, 버퍼로 전달되는 데이터를 필요에 따라 가공하거나, 화면에 출력할 수 있는 파일입니다.

1. RawToBmp : Raw 데이터를 Bmp포맷으로 변경

BYTE *pRAWBuffer : Raw 데이터가 저장된 버퍼의 포인터

BYTE *pBMPBuffer : Bmp 데이터를 저장할 버퍼의 포인터

int width : 영상의 폭

int height : 영상의 높이

2. CSampleGrabberCB::BufferCB : 영상데이터가 들어왔다는 이벤트를 받는 함수

double SampleTime : 시간

BYTE *pBuffer : 영상데이터 버퍼 포인터

long lBufferLen : 영상데이터의 길이

3. DisplayImage : 영상을 화면에 출력하는 함수

HDC hDC : 영상을 출력할 화면의 핸들

4. DisplayImage2 : 영상을 화면의 크기에 맞추어서 출력하는 함수

HDC hDC : 영상을 출력할 화면의 핸들

int ImageXpos : 화면의 폭

int ImageYpos : 화면의 높이

5. SaveImageToBmpFile : Bmp파일 저장함수

6. SAMPLE Source 설명

6. DisplayImageToScreen : 영상데이터를 화면크기에 맞추어서 출력

HWND hWnd: 영상데이터를 출력할 윈도우 핸들

int16 img_xsize: 윈도우의 폭

int16 img_ysize: 윈도우의 높이

7. Display_Thread : 영상데이터를 화면에 출력하기 위한 스레드

7. 기타 문의 사항

☞ 지금까지 설명한 드라이버 설치방법 외, 문의사항이 있으시면 저희 웹 카페 [훈텍 제품 자료실] → [제품 문의] 에 글을 남겨 주시거나, 아래의 연락처로 문의 하시기 바랍니다.

본사 : 경기도 성남시 중원구 상대원동 190-1 SK@테크노파크 테크동 810호

Phone : 031-776-3897, 3899

Fax : 031-776-3898

Homepage : <http://www.hontek.co.kr>

E-Mail : hontek@hontek.co.kr