

HTC-2300CB Camera

HTC-2300CB Camera Development Manual

Ver 2.1

2013. 06. 25



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

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

Jun 25, 2013

목 차

1. 개 요	4
2. HTC-2300CB Camera Specification	5
2.1 General feature	5
2.2 PC Requirements	5
2.3 Contents	6
2.4 Dimension and Description	6
2.5 Camera Interface	7
2.6 Trigger / Strobe Connector Port	7
2.7 Electrical Operating Condition	8
3. 개발요구사항	9
3.1 개발 요구사항	9
3.2 Development Architecture	9
4. Software Architecture	10
4.1 Multi Camera connection and Access	10
4.2 Sequence of Initialization	11
4.3 Main Control Dialog	12
4.4 System File description	24
4.5 ISP Control Dialog	25
4.5 OpenCV Control Dialog	28
5. API Lists	34
5.1 API Access Flow	34
5.2 Camera initialization, control command and image acquisition	35
5.3 ISP(Image Signal Processing) property and access	36

목 차

6. API 설명	37
6.1 HVR_camDrvInit	37
6.2 HVR_camSensorInit	38
6.3 HVR_camRegWrite	39
6.4 HVR_camRegRead	40
6.5 HVR_camRLEDOnOff	41
6.6 HVR_camGetImageData	42
6.7 HVR_ispResolutionCmd	43
6.8 HVR_ispResolutionCmdfotAOI	44
6.9 HVR_ispMoveStartPosOnAOI	45
6.10 HVR_ispVBlankAdjustment	46
6.11 HVR_ispVBlankAdjustmentRange	47
6.12 HVR_ispGetExposureRange	48
6.13 HVR_ispSetExposureTime	49
6.14 HVR_ispGetAWBGainControlRange	50
6.15 HVR_ispSetAWBGainControl	51
6.16 HVR_ispRawToBmp	54

1. 개요

본 문서는 HTC-2300CB 비전카메라를 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.

훈 텍

주소 : 경기도 성남시 중원구 상대원동 190-1 SK[®]테크노파크 테크동 810호

전화 : 031-776-3897,9

팩스 : 031-776-3898

홈페이지 : www.hontek.co.kr

2. HTC-2300CB Camera Specification

2.1 General feature

☞ 300만 화소(QXGA) 비전카메라	
☞ 1 / 2 Inch CMOS Sensor	
☞ Effective pixel	2048(H) x 1536(V)
☞ Frame rate	Max. 8fps@QXGA / Max. 30fps@VGA
☞ PC Interface	USB 2.0 (480Mbps)
☞ Sensor video data format	8Bit Bayer
☞ Sensor command interface	표준 I2C protocol
☞ Compression	압축 안 함
☞ Image Processing	Host 프로그램 구현 또는 센서 ISP 컨트롤 가능
☞ Multi-Camera connect	동일 PC 최대 3대 연결 가능

2.2 PC Requirements

☞ CPU	Pentium4 2.0GHz이상 (2.8GHz 이상 추천)
☞ RAM	512MB 이상 (1GB이상 권장)
☞ USB 2.0 Host Controller	Intel Controller chip 권장
☞ Operating System	Windows 2000 SP4 이상 / WindowsXP SP2이상

2. HTC-2300CB Camera Specification

2.3 Contents

☞ HTC-2300CB Camera

☞ Mini USB Cable(5Pin)

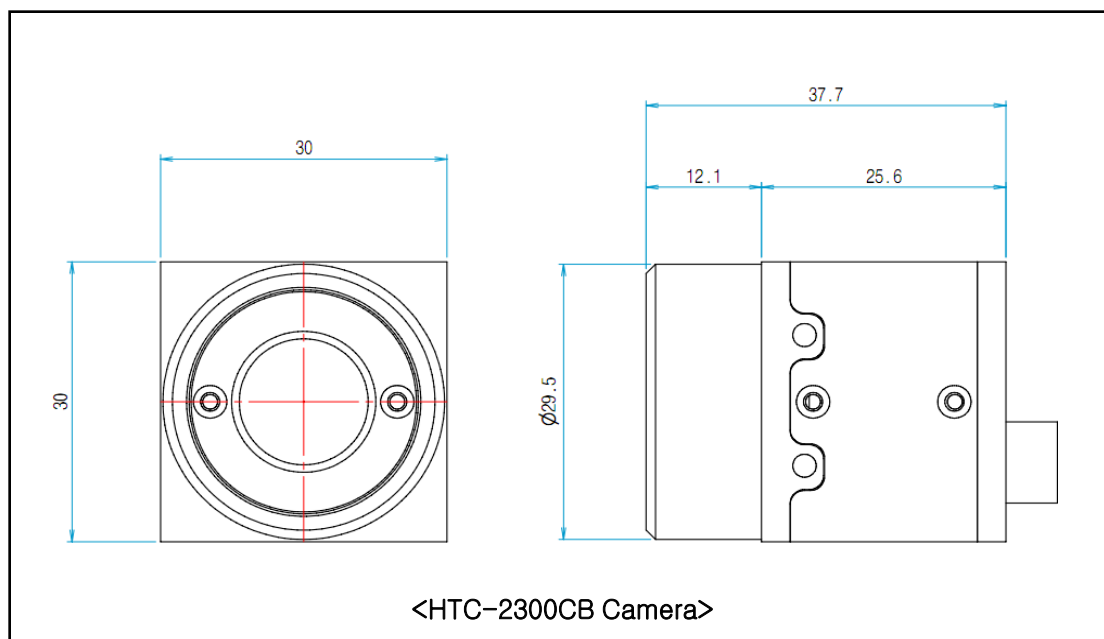
☞ User's manual / Program CD

※ Trig/Strobe 원형 Cable(4Pin) → (Option)



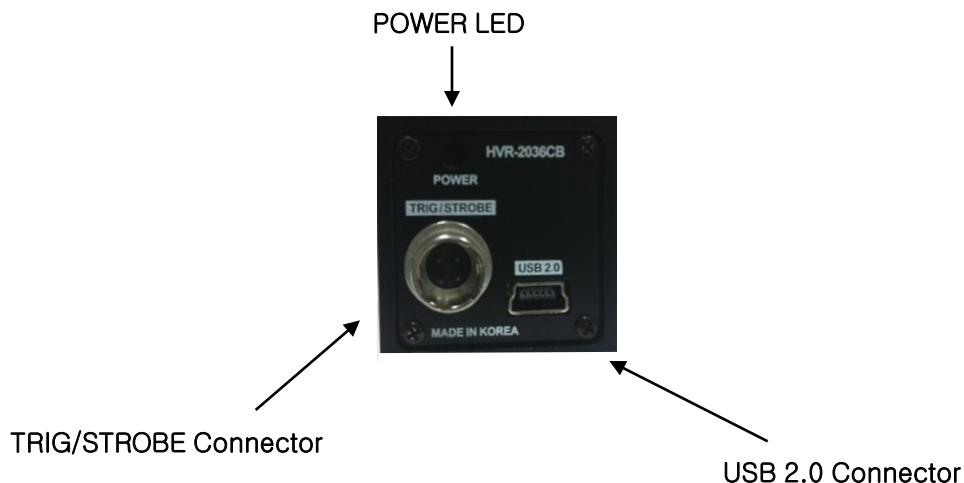
2.4 Dimension and Description

☞ Camera Body Size : 30(w) x 30(h) x 38(D) mm

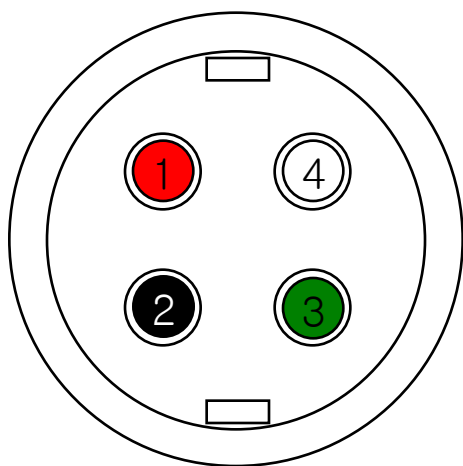


2. HTC-2300CB Camera Specification

2.5 Camera Interface



2.6 Trigger / Strobe Connector Port



<Camera Side>

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

* Wire Color

1 : Red

2 : Black

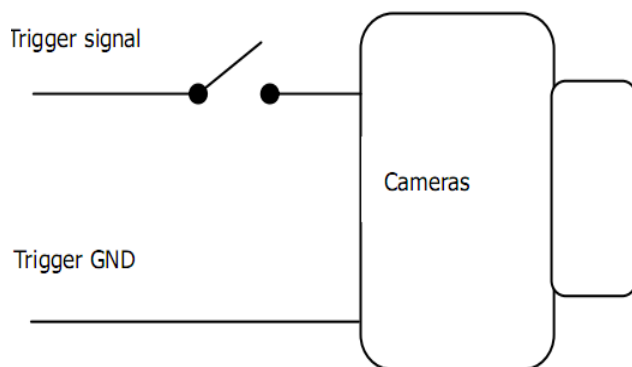
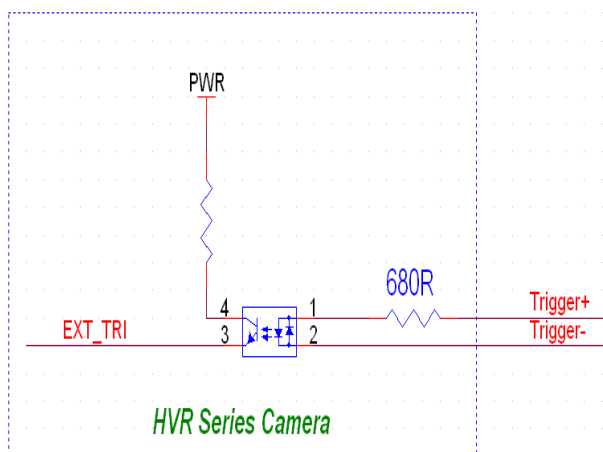
3 : Green

4 : White

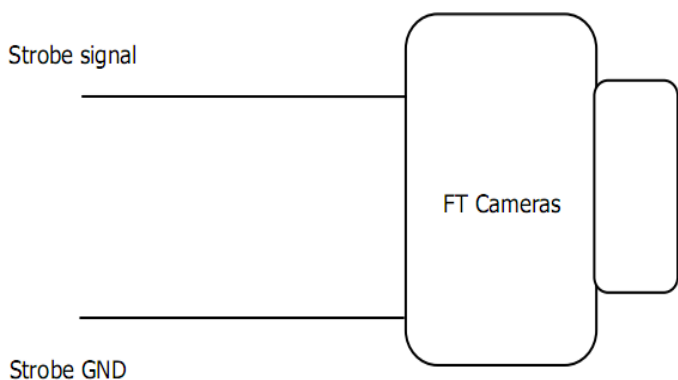
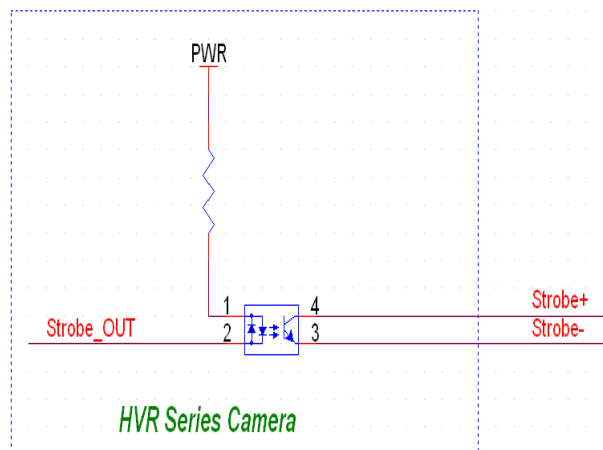
2. HTC-2300CB Camera Specification

2.7 Electrical Operating Condition

<Trigger>



<Strobe>

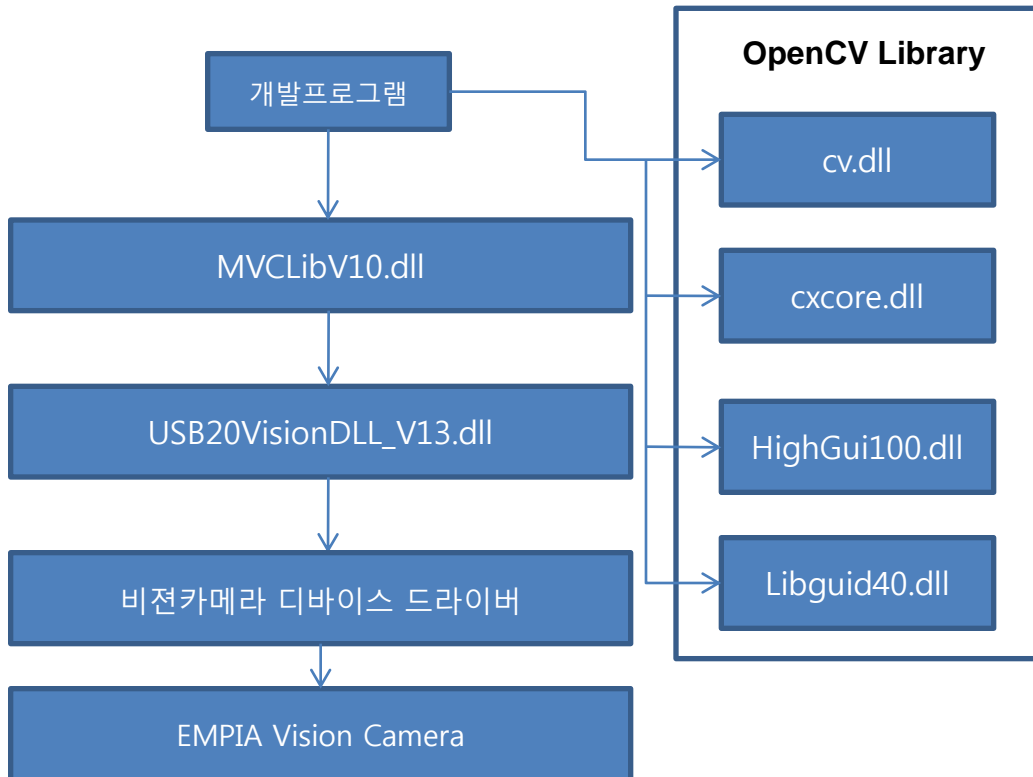


3. 개발 환경

3.1 개발 요구사항

- ☞ 운영체제 : Microsoft Windows 2000 SP4 이상, WindowsXP SP2 이상,
- ☞ 컴파일러 : Microsoft Visual Basic / Visual C++ 6.0
Visual Basic .NET / Visual C++ .NET
- ☞ 기타 : openCV 라이브러리

3.2 Development Architecture



4. Software Architecture

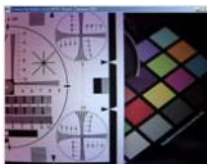
4.1 Multi Camera connection and Access

HTC-2300CB 카메라는 동일 PC상에 최대 3대의 카메라 연결이 가능합니다. 카메라에 시리얼 번호를 각각 부여하여(1001~1003) 각 카메라를 식별합니다.

Single Connection



Serial_Number[5]=L"1001";

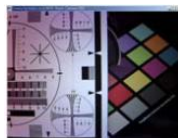


M fps

Multi Connection



Serial_Number[5]=L"1001";



M/N fps



Serial_Number[5]=L"1002";



M/N fps



Serial_Number[5]=L"1003";



M/N fps

Fig 1. Multi Camera connection and Access

4. Software Architecture

4.2 Sequence of Initialization

실행파일 실행시, Device Check에 관한 초기화 과정을 설명합니다. 개발자 분은 아래 과정을 참조하시기 바랍니다.

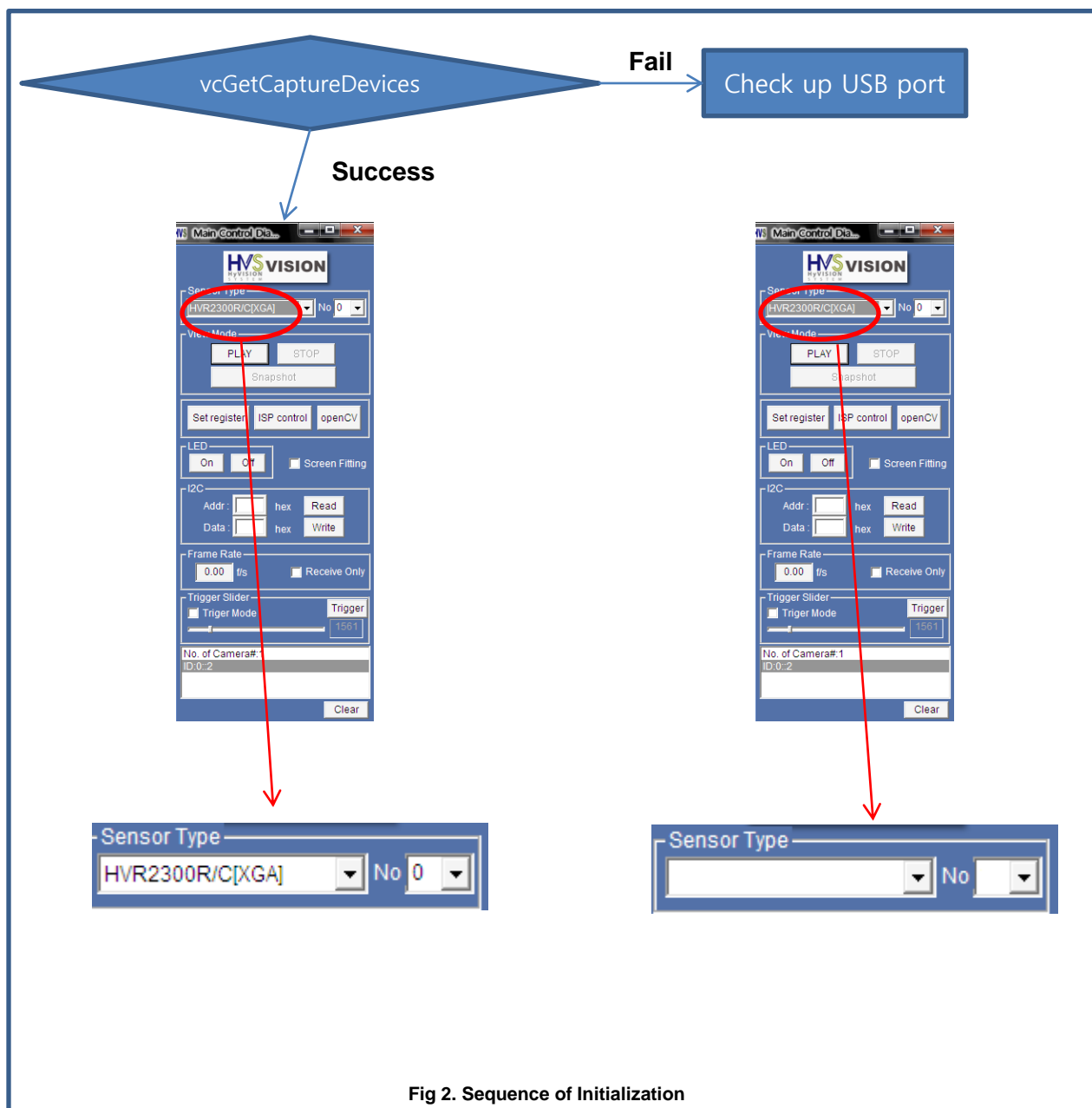
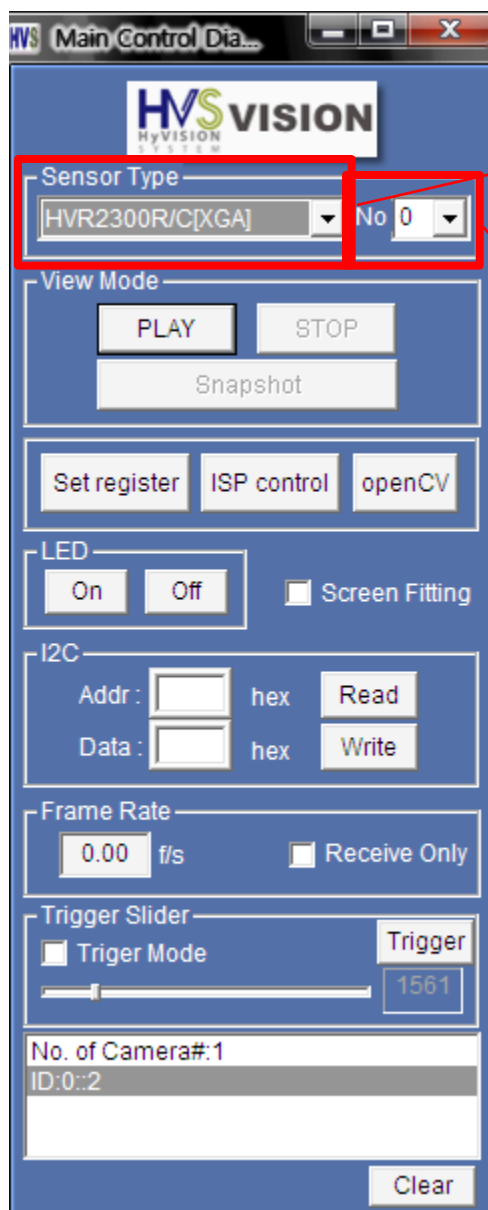


Fig 2. Sequence of Initialization

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.1 카메라 검색

Sensor Type 인식모델의 해상도 List를 보여준다.

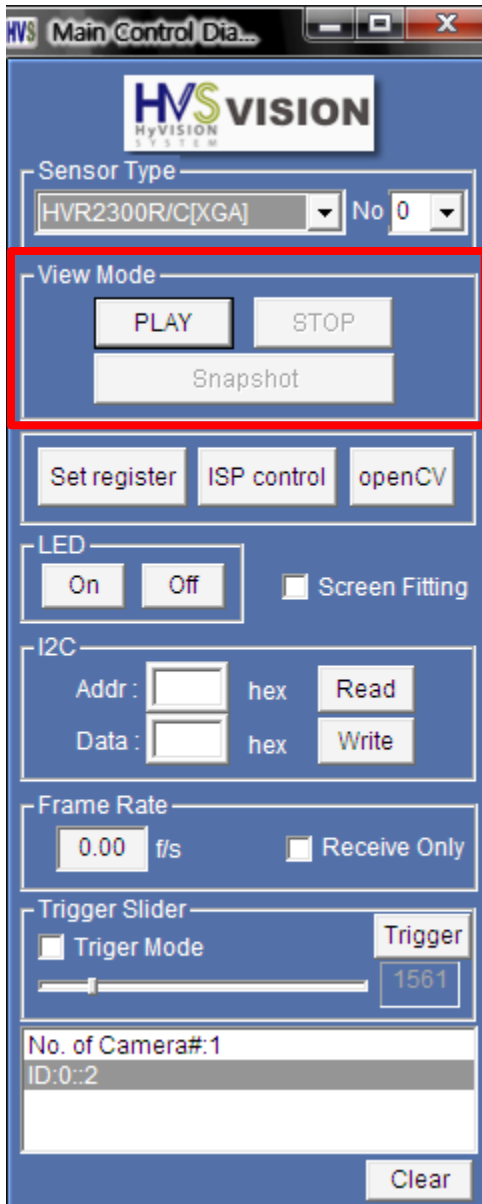
No 동일모델이 연결된 대수를 index로 표현한다.

Fig 3. Main Control Dialog - 해상도

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.2 View Mode

선택된 카메라를 실행시키거나 중지, 또는 정지영상, 동영상을 저장한다.

PLAY : 카메라를 실행

STOP : 카메라 실행을 중지

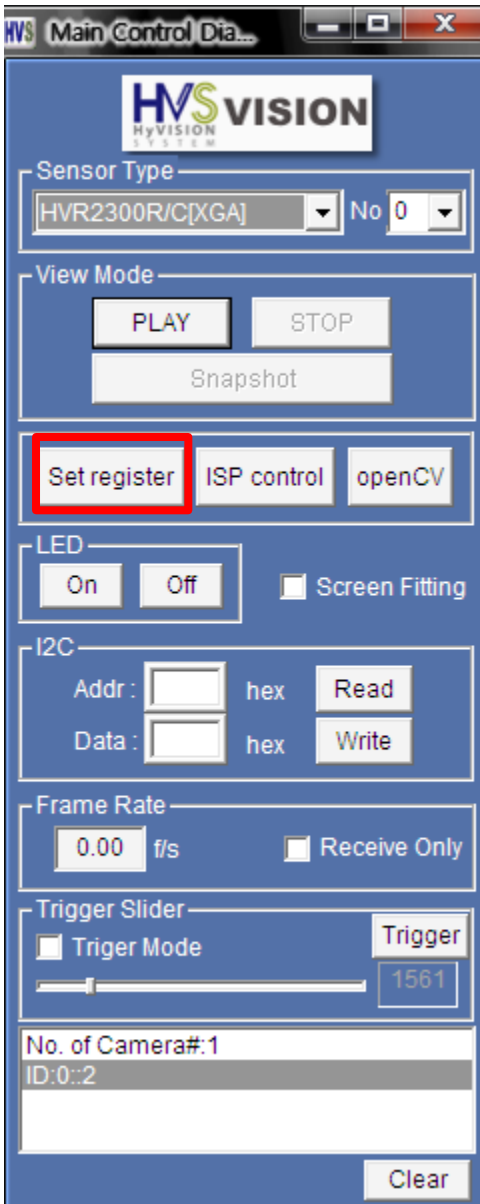
Snapshot : 카메라 실행 시에만 동작하며, 하나의 정지영상을 BMP파일로 저장한다.

Fig 4. Main Control Dialog - 해상도

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.3 Set Register

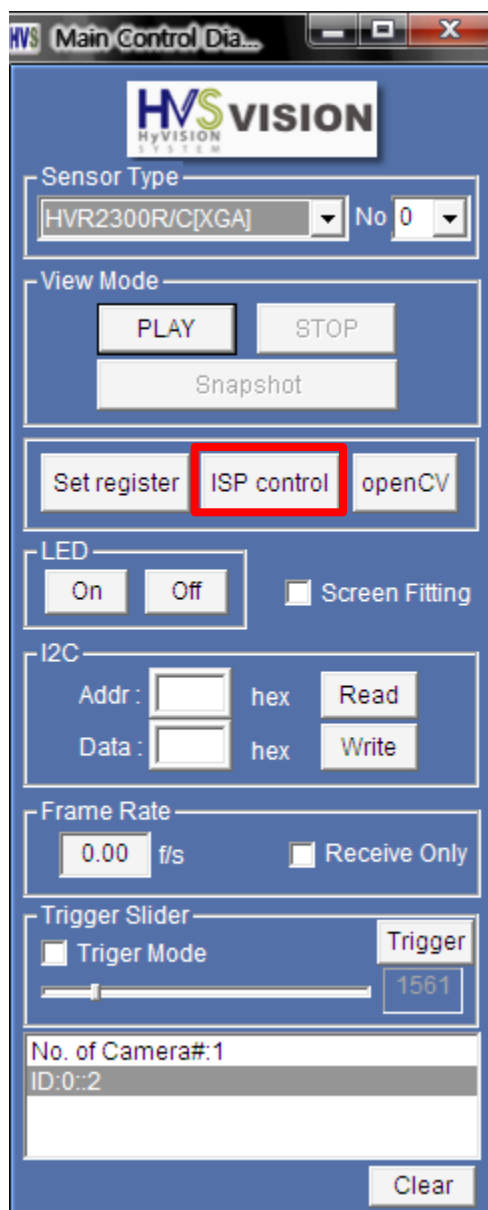
카메라에 입력된 추가 레지스터를 실행한다.

Fig 5. Main Control Dialog – ISP Control

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.4 ISP(Image Signal Processing) Control

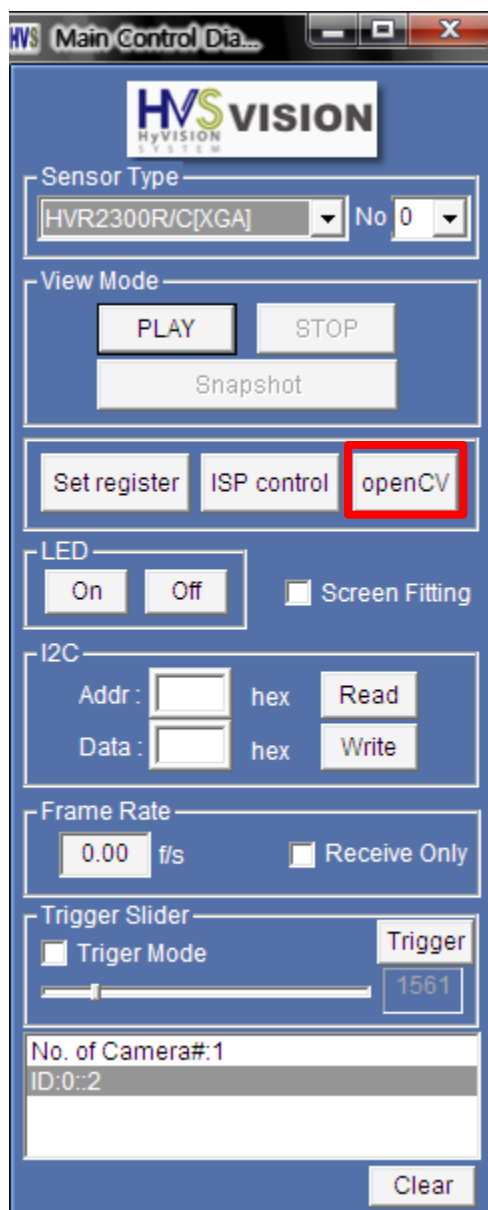
카메라로부터 입력되는 영상을 가공하는 기능을 담당

Fig 5. Main Control Dialog – ISP Control

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.5 OpenCV

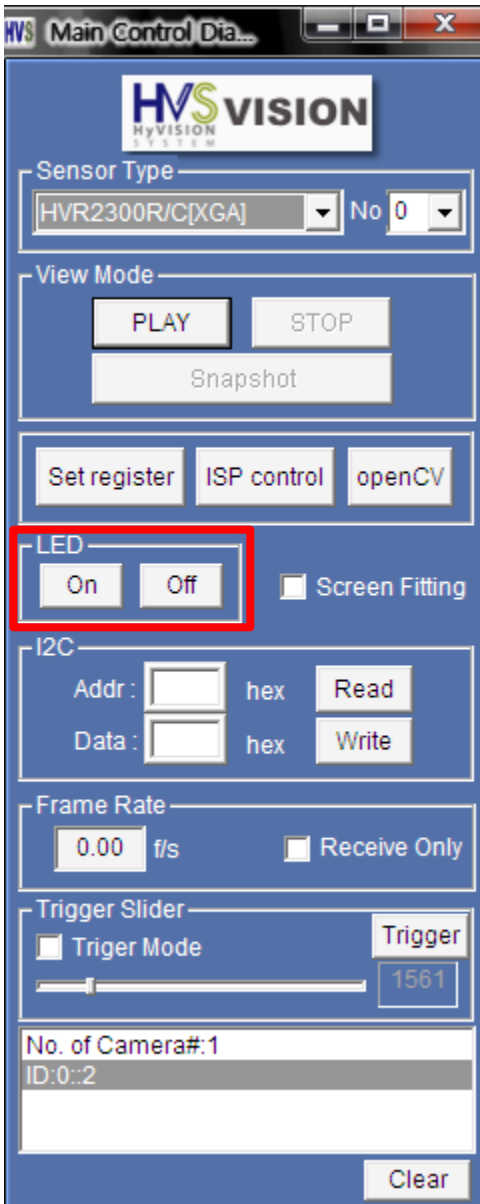
OpenCV라이브러리의 이용하여 영상 데이터에 기본적인 효과를 적용

Fig 6. Main Control Dialog - OpenCV

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.6 LED On/Off

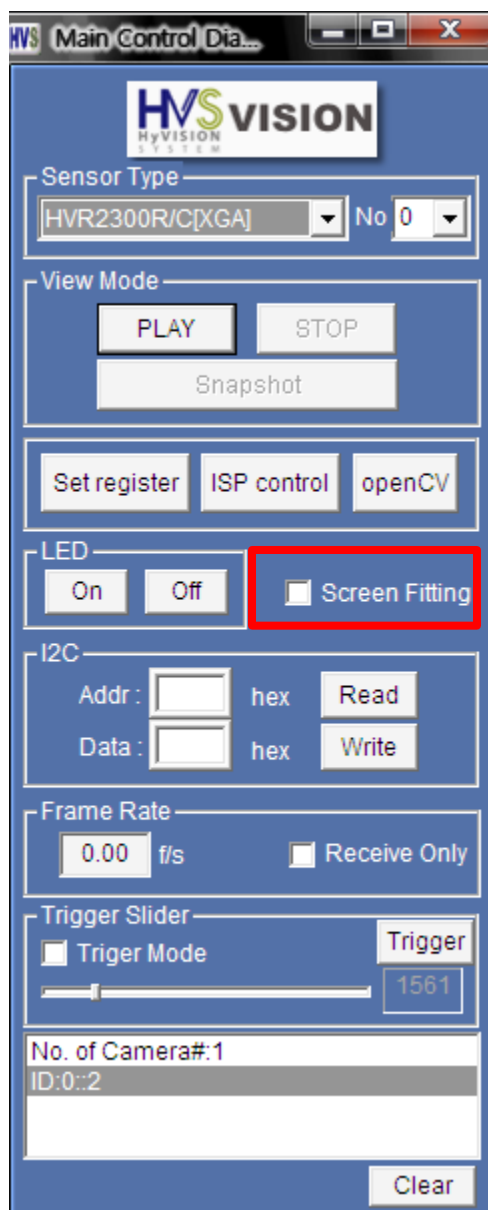
카메라 외부에 연결된 LED Light를 컨트롤 한다.

Fig 6. Main Control Dialog - OpenCV

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.7 Screen Fitting

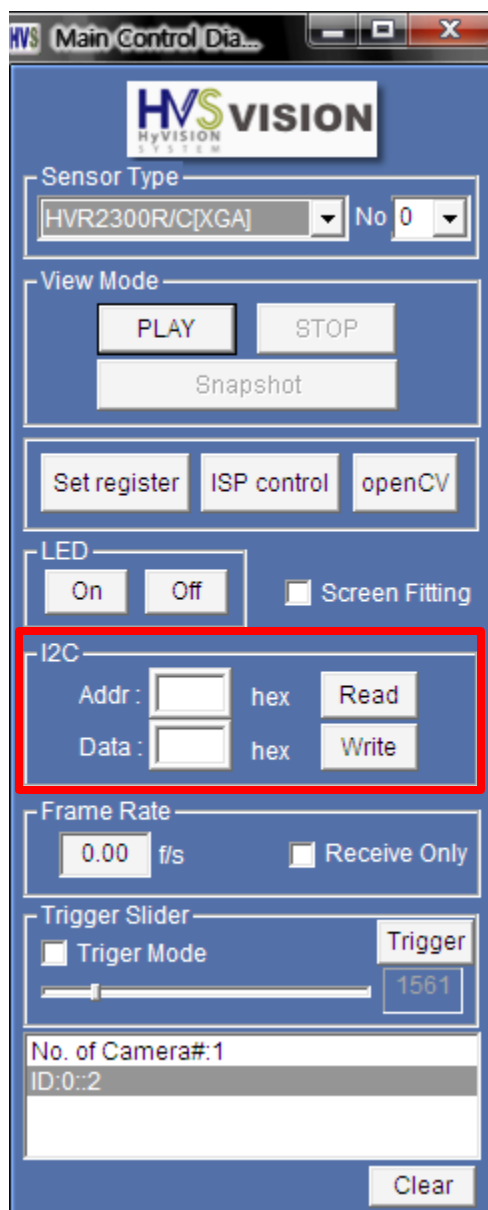
출력 영상의 해상도가 사용자 모니터 해상도보다 클 경우, 모니터 해상도에 맞게 출력한다.

Fig 6. Main Control Dialog - OpenCV

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.8 I2C Read/Write

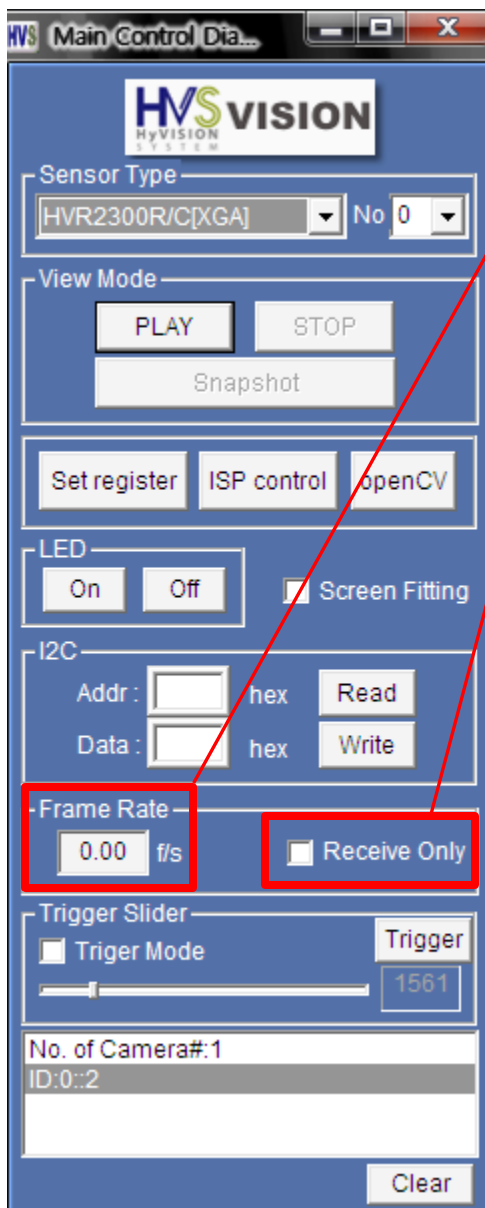
표준 I2C Command 방식의 입/출력을 실행한다.

Fig 6. Main Control Dialog - OpenCV

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.9 Frame Rate

1초당 출력하는 프레임 수

4.3.10 Receive Only

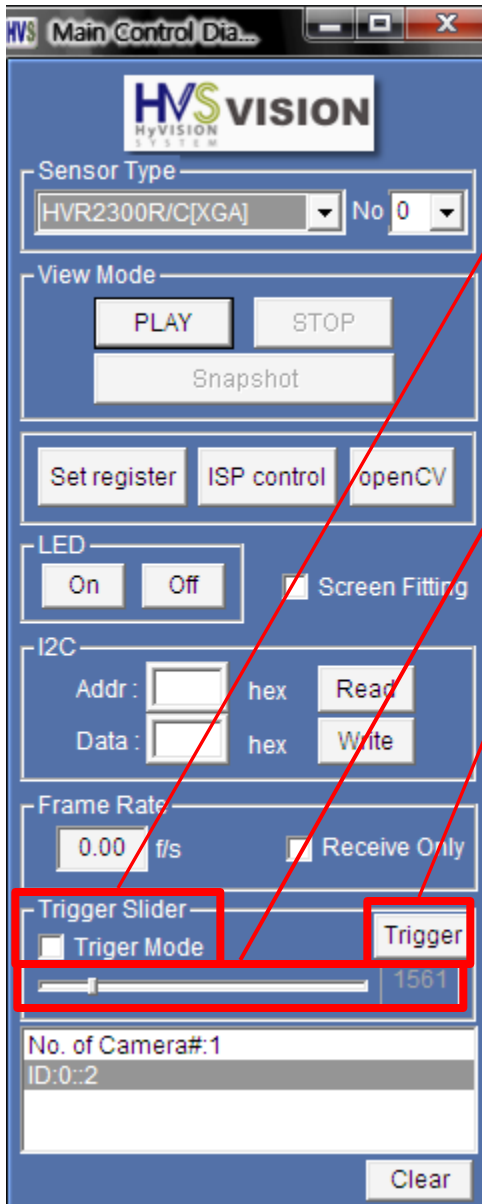
데이터를 가져오지만 출력은 하지 않는다.

Fig 6. Main Control Dialog - OpenCV

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.11 트리거

Triger Mode : 트리거 모드 On/Off
On: HVR_camRegWrite(gProductID, gDrvIndex, 0x1e, 0xc740);
Off: HVR_camRegWrite(gProductID, gDrvIndex, 0x1e, 0xc040);

트리거 슬라이더 바 : 노출 설정
HVR_camRegWrite(gProductID, gDrvIndex, 0x09, gSlider);

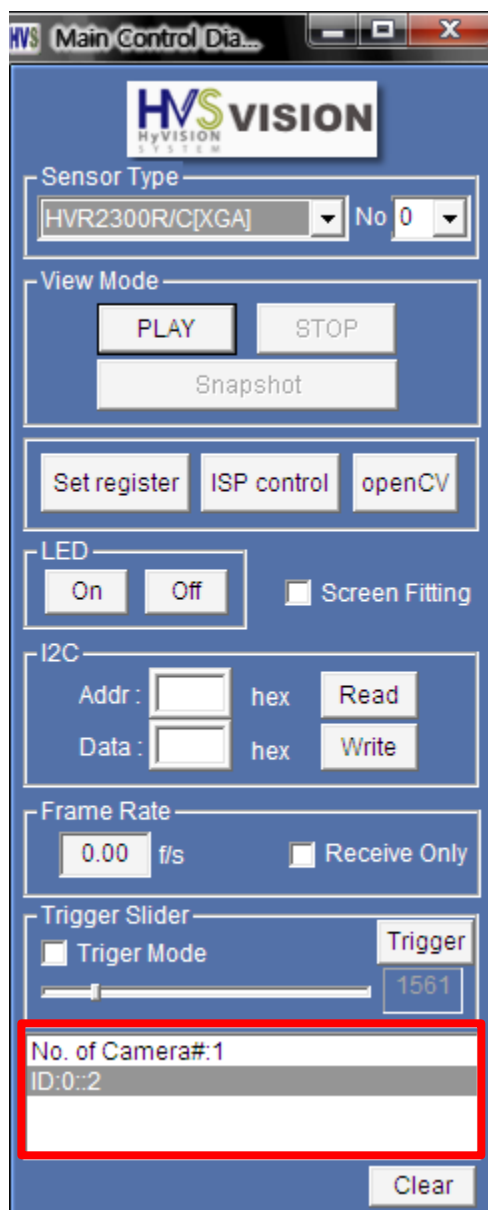
Triger 버튼 : 트리거 사용 버튼
HVR_camRegWrite(gProductID, gDrvIndex, 0xb, 0x1);

Fig 6. Main Control Dialog - OpenCV

4. Software Architecture

4.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



4.3.12 상태정보

카메라의 상태나, 실행정보 등을 출력

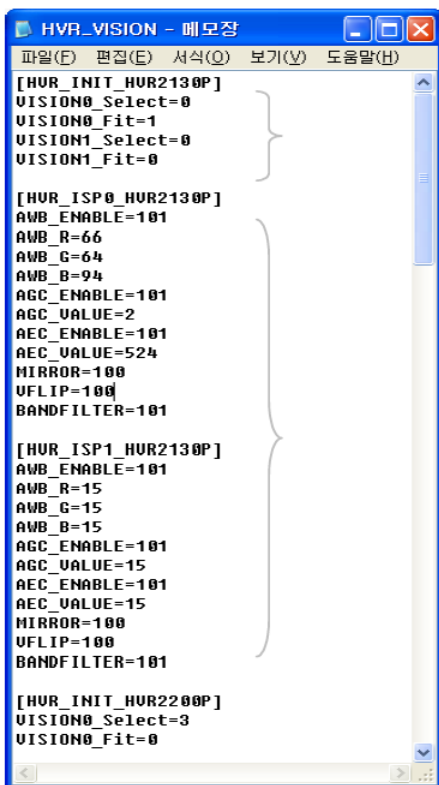
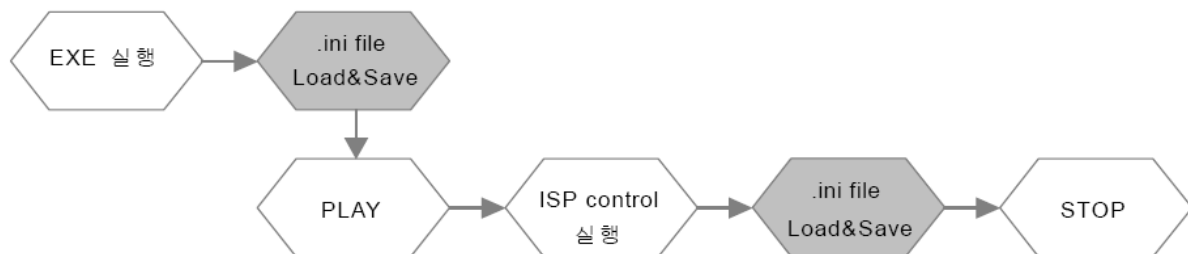
Fig 11. Main Control Dialog - 상태정보창

4. Software Architecture

4.4 System File description

[1] HVR_VISION.ini File

ISP를 설정할 수 있는 다이얼로그



[HVR_INIT_xxxxxxx] 제품모델별로 구성됨.

VISION_i_Select=0 [SensorType]에 대한 초기값.

VISION_i_Fit=1 [Preview Fit]에 대한 초기값.

'i'은 [Cam's #]와 동일한 index를 가진다.

[HVR_ISP_i_xxxxxxx] 제품모델별로 구성됨.

'i'은 [Cam's #]와 동일한 index를 가진다.

AWB_ENABLE=vvv [ISP control]의 AWB on/off 초기값.

AWB_R=vvv [ISP control]의 Red gain 초기값.

AWB_G=vvv [ISP control]의 Green gain 초기값.

AWB_B=vvv [ISP control]의 Blue gain 초기값.

AGC_ENABLE=vvv [ISP control]의 AGC on/off 초기값.

AGC_VALUE=vvv [ISP control]의 AGC manual 초기값.

AEC_ENABLE=vvv [ISP control]의 AEC on/off 초기값.

AEC_VALUE=vvv [ISP control]의 AEC manual 초기값.

MIRROR=vvv [ISP control]의 Mirror 초기값.

VFLIP=vvv [ISP control]의 V-flip 초기값.

BANDFILTER=vvv [ISP control]의 Band Filter 선택 초기값.

'vvv'은 ISP 항목별 설정값을 가진다.

Fig 12. Soft ISP Control Dialog - 화이트 발란스

4. Software Architecture

4.4 System File description

[2] Example.set File

영상 Play 시 센서에게 보내질 사용자 임의 Register Command 들을, 아래의 .set파일 형태로 제작한 후 Load하여 사용하는 파일이다. 센서 컨트롤과 관련해 특이사항 없는 한 사용하지 않는 것이 좋다.

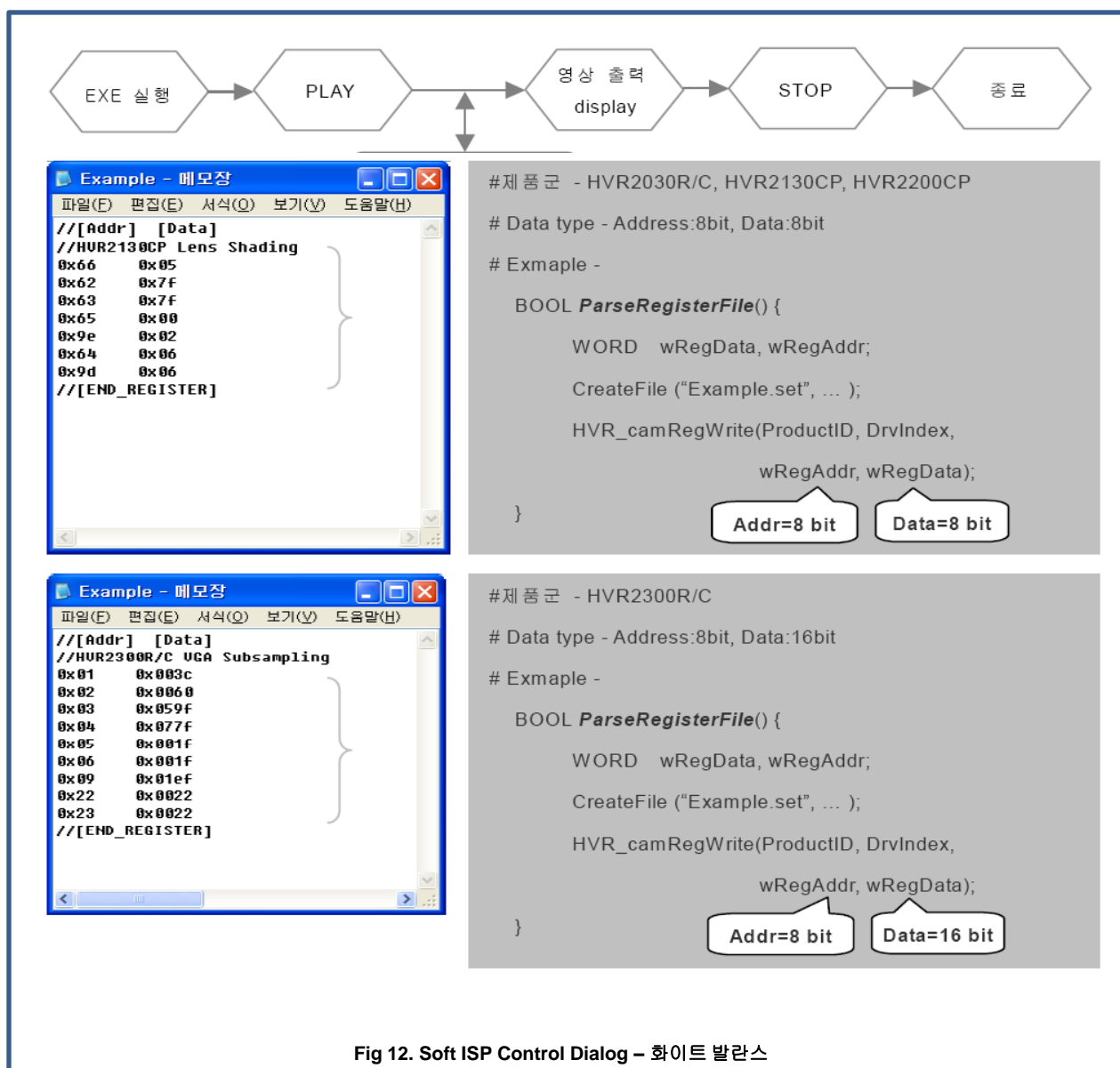


Fig 12. Soft ISP Control Dialog – 화이트 발란스

4. Software Architecture

4.5 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

2.5.1 Gain Control

카메라로부터 입력된 영상의 RGB GAIN에 대한 가중치를 설정

Min : 0

Max : 183

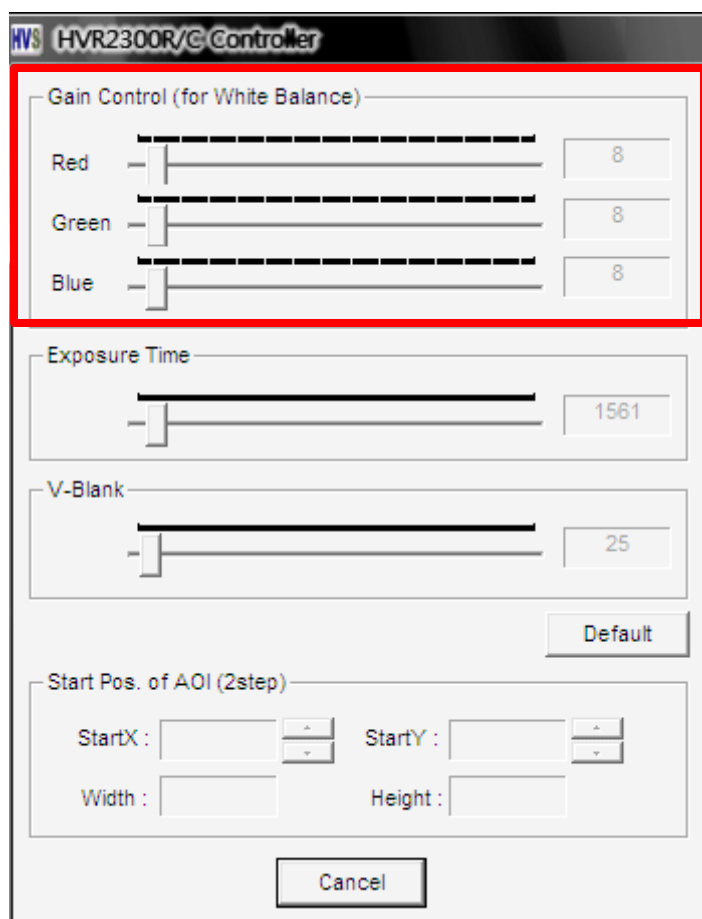


Fig 12. Soft ISP Control Dialog – 화이트 발란스

4. Software Architecture

4.5 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

2.5.2 Exposure Time

카메라로부터 입력된 영상의 ExposureTime(노출 시간)을 설정한다.

Min : 0

Max : 32767

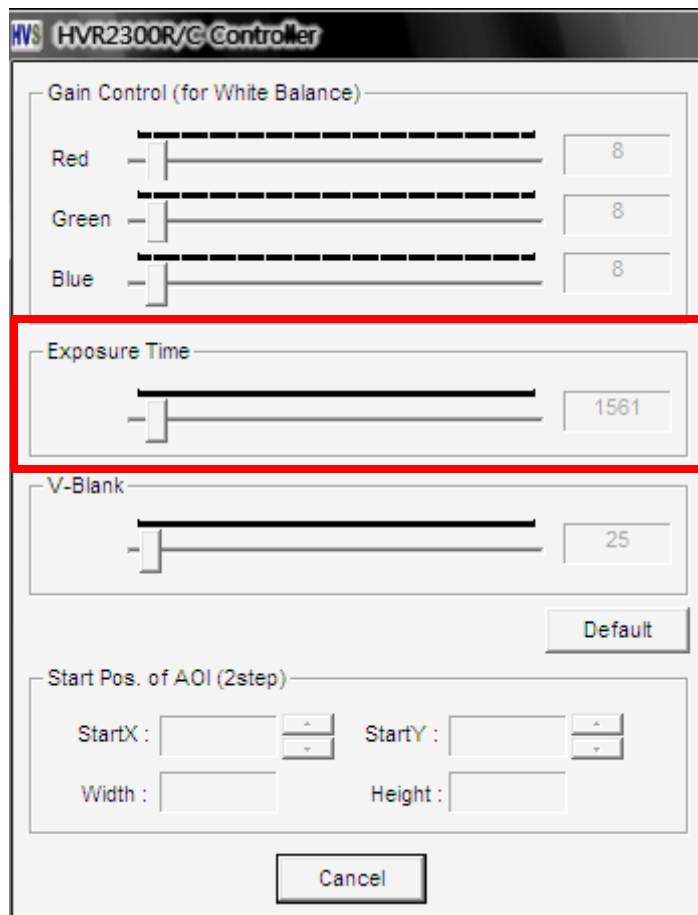


Fig 12. Soft ISP Control Dialog – 화이트 발란스

4. Software Architecture

4.5 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

2.5.3 V-Blank

카메라 영상의 프레임간 간격을 조정한다..

Min : 0

Max : 767

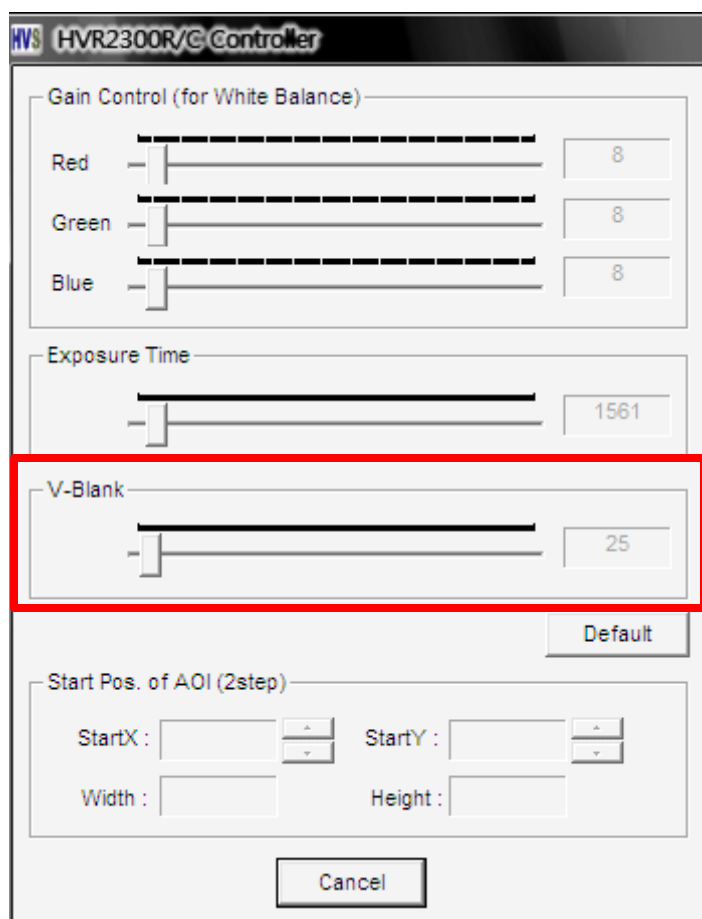
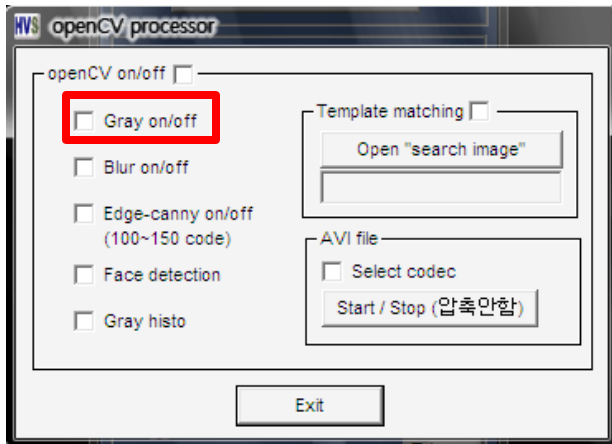


Fig 12. Soft ISP Control Dialog – 화이트 발란스

4. Software Architecture

4.6 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



4.6.2 Gray

영상을 회색영상으로 보여준다.

```
CVAPI(void) cvCvtColor(
    const CvArr* src,
    CvArr* dst,
    int code );
```

Src : 원본영상

Dst : 출력영상

Code : 변형타입(BGR2GRAY)

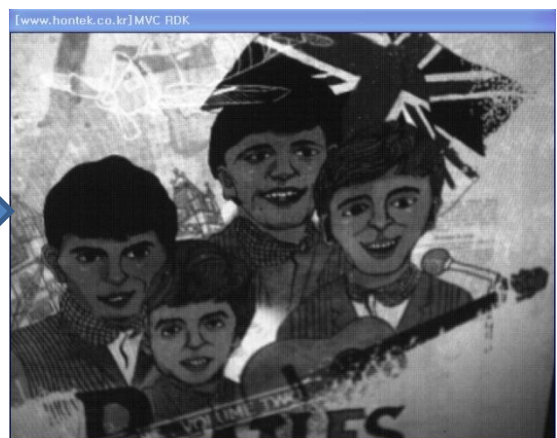
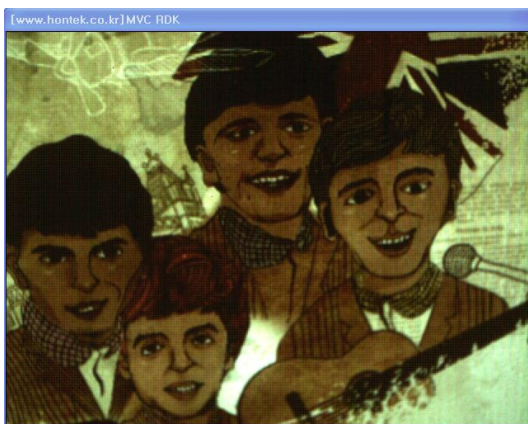
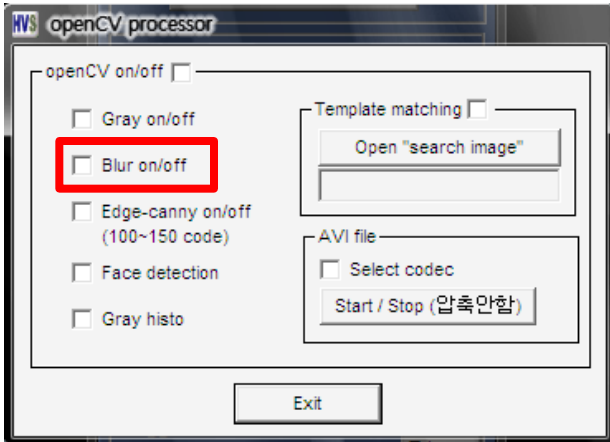


Fig 15. OpenCV Control Dialog - Gray

4. Software Architecture

4.6 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



4.6.3 Blur

영상을 흐리게 나타낸다.

```
CVAPI(void) cvSmooth(
    const CvArr* src, // 입력 영상
    CvArr* dst, // 출력 영상
    int smoothtype, // 블러링 타입
    int param1, // 영상 너비
    int param2, // 영상 높이
    );
```

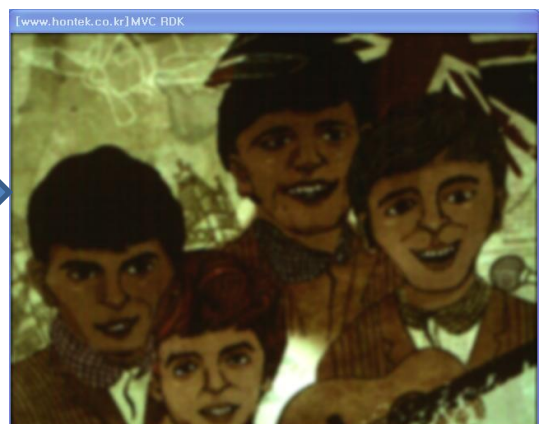
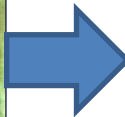
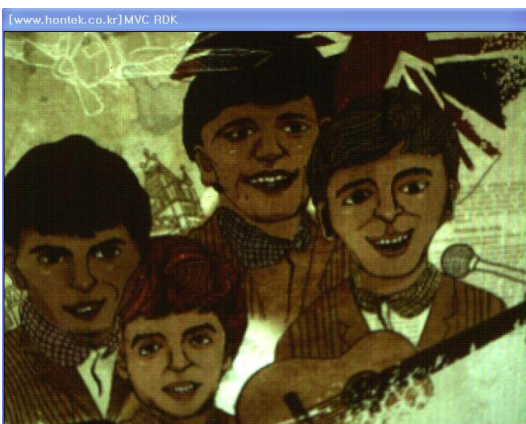
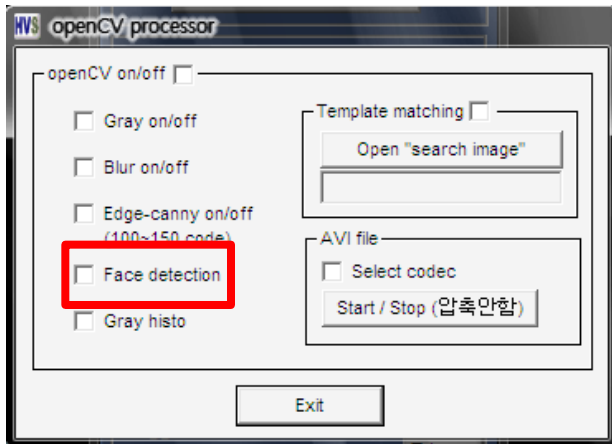


Fig 16. OpenCV Control Dialog - Blur

4. Software Architecture

4.6 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



4.6.4 Face Detection

화면에서 얼굴을 찾아낸다.

```
IplImage *
MVC_OpenCV_FaceDetection(
IplImage* image);
```

Image : 원본 영상

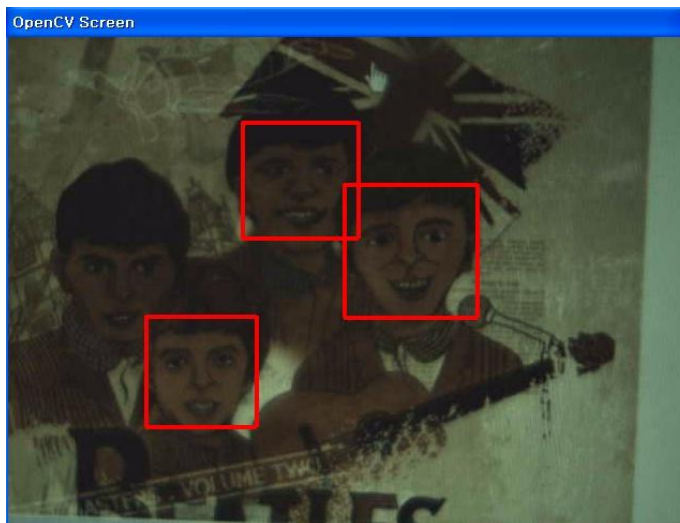
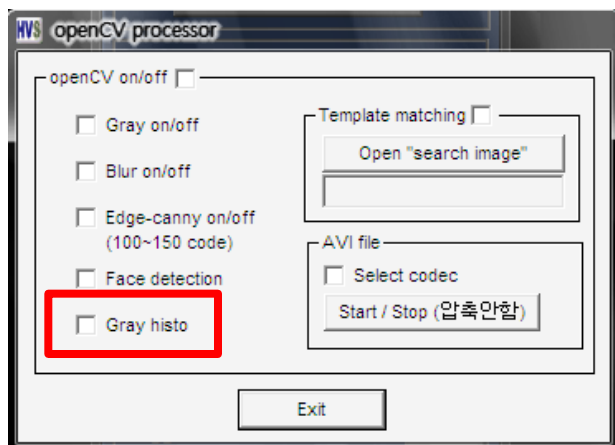


Fig 17. OpenCV Control Dialog – Face Detection

4. Software Architecture

4.6 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



4.6.5 Gray Histo

영상의 밝기분포를 보여준다.

```
CV_INLINE void cvCalcHist(
    IplImage** image,
    CvHistogram* hist,
    int accumulate CV_DEFAULT(0),
    const CvArr* mask
    CV_DEFAULT(NULL) )
```

Image : 원본데이터

Hist : 히스토그램 데이터 구조체

Accumulate : 각 빈도 누적 여부

Mask : 마스크 영상

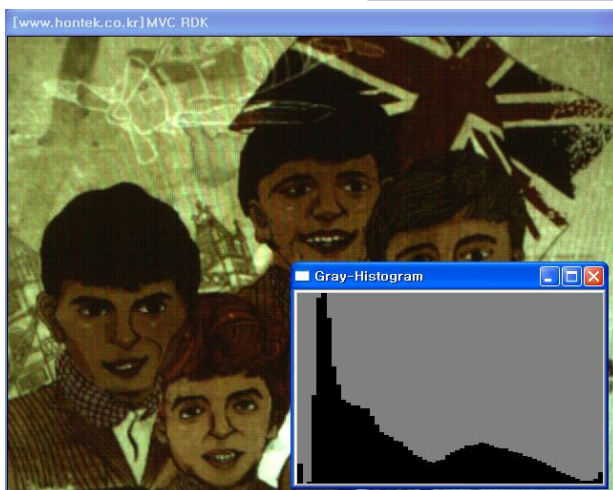
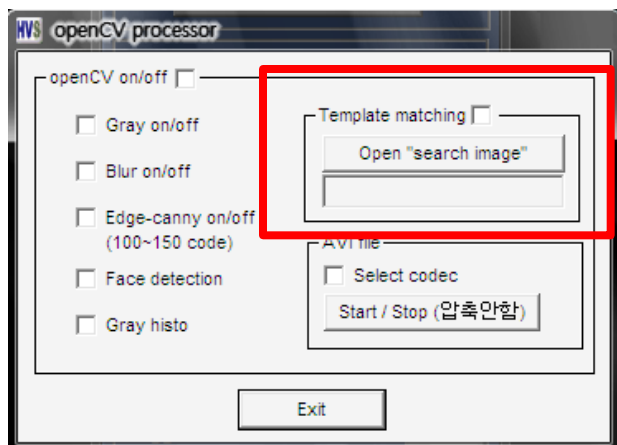


Fig 18. OpenCV Control Dialog – Gray Histo

4. Software Architecture

4.6 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



4.6.6 Template Matching

영상에서 특정영상을 찾는다.

```
CVAPI(void) cvMatchTemplate(
const CvArr* image,
const CvArr* templ,
CvArr* result,
int method );
```

Image : 영상 데이터

Templ : 검색할 영상

Result : 결과 영상

Method : 검출방법

CV_TM_SQDIFF 0

CV_TM_SQDIFF_NORMED 1

CV_TM_CCORR 2

CV_TM_CCORR_NORMED 3

CV_TM_CCOEFF 4

CV_TM_CCOEFF_NORMED 5

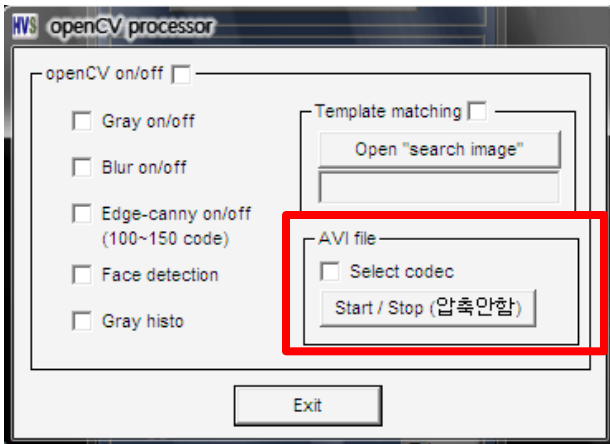


Fig 22. OpenCV Control Dialog – Template Matching

4. Software Architecture

4.6 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



4.6.7 동영상 저장

동영상으로 저장한다.

```
// 저장시작
VideoOut =
cvCreateVideoWriter(
    tmp,
    -1,
    value-5,
    cvGetSize(imgGray),
    1);
// 저장
cvWriteFrame(VideoOut, imgIntP);
// 저장 종료
cvReleaseVideoWriter(&VideoOut);
```

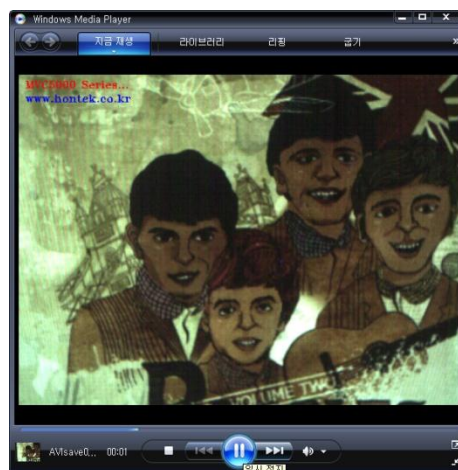


Fig 23. OpenCV Control Dialog – 동영상 저장

5. API Lists

5.1 API Access Flow

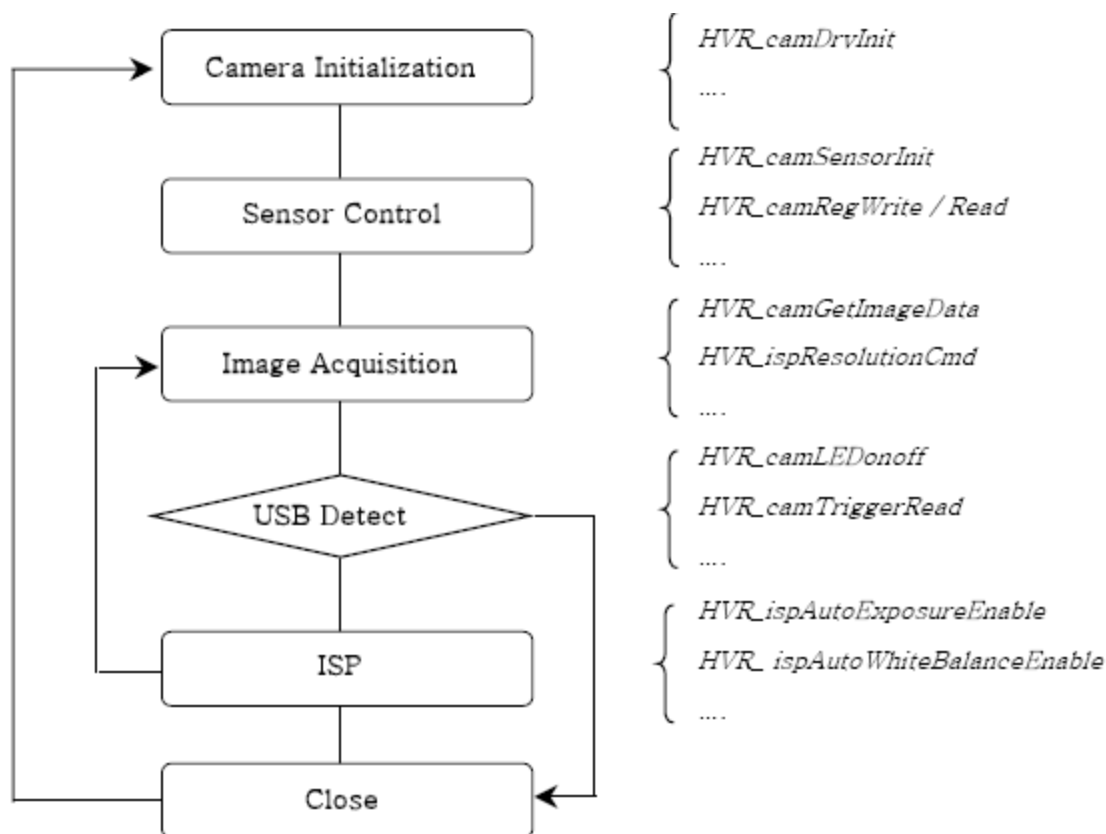


그림 4 : API access Flow

5. API Lists

5.2 Camera initialization, control command and image acquisition

API	내용
HVR_camDrvInit	연결된 Camera를 초기화하고 식별자를 구한다.
HVR_camSensorInit	Camera가 사용하는 센서를 초기화 한다.
HVR_camRegWrite	센서를 직접 컨트롤 한다.(쓰기 명령)
HVR_camRegRead	센서를 직접 컨트롤 한다.(읽기 명령)
HVR_camLEDonoff	LED 신호를 컨트롤 한다.
HVR_camGetImageData	영상 데이터를 출력한다.

5. API Lists

5.3 ISP(Image Signal Processing)property and access

API	내용
HVR_ispResolutionCmd	영상사이즈를 지정한다.
HVR_ispResolutionCmdforAOI	AOI(Area Of Interest)영역을 초기화한다.
HVR_ispMoveStartPosOnAOI	설정된 영상의 Frame간 간격을 조절한다.
HVR_ispVBlankAdjustment	카메라 영상의 Frame간 간격을 조절한다.
HVR_ispVBlankAdjustmentRange	카메라 영상의 Frame간 간격값의 범위를 얻는다
HVR_ispSetExposureTime	Exposure Time 을 Manual 조절한다..
HVR_ispGetExposureRange	Exposure Time의 Range 와 current 값을 얻는다.
HVR_ispSetAWBGainControl	R/G/B 각각의 Gain을 컨트롤 한다.
HVR_ispGetAWBGainControlRange	R/G/B 각각의 Gain range와 현재값을 얻는다.
HVR_ispRawToBmp	RAW포맷의 데이터를 BMP포맷으로 전환한다.

6. API 설명

6.1 HVR_camDrvInit

연결된 Camera를 초기화하고 해당 카메라의 식별자를 구한다.

구분	내용	
Syntax	Int HVR_camDrvInit (int *DrvInfo)	
Parameter	Drvinfo	1개 이상의 카메라를 구별하기 위해 고유 식별자를 얻어온다.
Return Value	카메라의 연결 대수를 리턴한다. 따라서 0일 경우 카메라가 연결되지 않았음을 의미한다.	
Sample	<pre> Int nDrvNum; Int nDrvInform[MAX_DEVICE] Memset(nDrvInform, NULL, sizeof(nDrvInform)); nDrvNum = HVR_camDrvInit(&nDrvInform[0]); For(int l = 0; l < MAX_DEVICE; i++) printf("device sequence=%d, id=%x",l,nDrvInform[i]); </pre>	

6. API 설명

6.2 HVR_camSensorInit

Camera가 사용하는 센서를 초기화 한다.

구분	내용	
Syntax	BOOL HVR_camSensorInit(int ProductNo, int DeviceNo);	
Parameter	ProductNo	사용 제품의 모델명을 입력한다..
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않았음을 의미	
Sample	<pre> BOOL re; Re = HVR_camSensorInit(HVR2300R,0); If(!re) return FALSE; </pre>	

6. API 설명

6.3 HVR_camRegWrite

Camera의 센서부를 직접 컨트롤 한다(쓰기 명령)

(단, 제품 특성을 이해하지 않고 사용시 오동작을 일으킬 수 있으니, 가능한 사용을 금한다)

구분	내용	
Syntax	BOOL HVR_camRegWrite(int ProductNo, int DeviceNo, WORD wRegAddr, WORD wRegData);	
Parameter	ProductNo	사용 제품의 모델명을 입력한다..
	DeviceNo	카메라의 순번을 입력한다.
	wRegAddr	센서의 해당 레지스터 번지를 입력한다.
	wRegData	센서에게 보내질 명령 data를 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않았음을 의미	
Sample	BOOL re; Re = HVR_camRegWrite(HVR2300R,0, 0x00, 0xff); If(!re) return FALSE;	

6. API 설명

6.4 HVR_camRegRead

Camera의 센서부를 직접 컨트롤 한다(읽기 명령)

(단, 제품 특성을 이해하지 않고 사용시 오동작을 일으킬 수 있으니, 가능한 사용을 금한다)

구분	내용	
Syntax	WORD HVR_camRegRead(int ProductNo, int DeviceNo, WORD wRegAddr);	
Parameter	ProductNo	사용 제품의 모델명을 입력한다..
	DeviceNo	카메라의 순번을 입력한다.
	wRegAddr	센서의 해당 레지스터 번지를 입력한다.
Return Value	센서로 부터 읽은 데이터	
Sample	WORD data; data = HVR_camRegWrite(HVR2300R,0, 0x00, 0xff); return data;	

6. API 설명

6.5 HVR_camRLEDOff

H/W에서 구성한 외부 LED 램프를 실시간 On/Off 하기 위한 함수로 사용된다.

(별도의 H/W 구성은 H/W parts manual을 참조한다)

구분	내용	
Syntax	BOOL HVR_camLEDOnoff(WORD bOnOff, int DeviceNo);	
Parameter	bOnOff	1이면 LED On, 0이면 LED Off
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상, 0이면 동작하지 않음을 의미한다.	
Sample	BOOL re, bOnOFF; If(bOnOff) re = HVR_camLEDOnoff(1,0); Else re = HVR_camLEDOnoff(0,0); If(!re) return FALSE;	

6. API 설명

6.6 HVR_camGetImageData

카메라 영상 data를 얻어오는 함수

구분	내용	
Syntax	<pre>Int HVR_camGetImageData(int ilmageLen, LPBYTE pRawBuffer, int DeviceNo);</pre>	
Parameter	ilmageLen	영상 data의 사이즈를 입력한다.
	pRawBuffer	영상 data를 담을 버퍼포인터.
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상이미지를 전달 -1이면 비정상 이미지가 전달 -2이면 영상출력 중 해당 카메라의 USB포트가 분리되었음을 전달	
Sample	<pre> BOOL re; Int cnt,reData; BYTE *pBuf; pBuf = new BYTE[640*480]; // 512배수 While(1){ Sleep(1); reData = HVR_camGetImageData(640*480,pBuf, 0); if(reData == USB_DATA_PASS) { // reData가 1일 경우만 출력 } else if(reData == USB_DATA_FAIL){ continue; } else if(reData == USB_BAND_HALT){ return false; } } Delete pBuf;</pre>	

6. API 설명

6.7 HVR_ispResolutionCmd

카메라에 대한 초기 설정이 끝난 후, display를 위한 해당 HDC에 출력 사이즈 만큼 미리보기가 시작된다.

구분	내용	
Syntax	Int HVR_ispResolutionCmd(int Resolution, int ProductNo, int DeviceNo);	
Parameter	Resolution	출력 해상도를 입력 한다.
	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 동작, 0이면 동작하지 않음을 의미	
Sample	<pre> BOOL re; If(640*480) re = HVR_ispResolutionCmd(VGA, HVR2300R,0); If(640*480 && bWindowing) re = HVR_ispResolutionCmd(Gaw, HVR2300R,0); If(!re) return FALSE; </pre>	

6. API 설명

6.8 HVR_ispResolutionCmdforAOI

카메라의 최대 지원해상도에서 일정 영역만큼 잘라내어 출력하기 위한 기능으로서, 시작점과 가로/세로 길이를 입력하여 초기화한다.

구분	내용	
Syntax	<pre>Int HVR_ispResolutionCmdforAOI(int ProductNo, int DeviceNo, int StartX, int StartY, int Width, int Height);</pre>	
Parameter	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
	StartX	가로 시작점
	StartY	세로 시작점
	Width	가로 길이
	Height	세로 길이
Return Value	1이면 동작, 0이면 동작하지 않음을 의미	
Sample	<pre>BOOL re; // 최대 지원 해상도 command Re = HVR_ispResolutionCmd(QXGA, HVR2300R, 0); If(bEnable) // HDTV re = HVR_ispResolutionCmdforAOI(HVR2300R, 0,0,0,1280,720); If(!re) return FALSE;</pre>	

6. API 설명

6.9 HVR_ispMoveStartPosOnAOI

HVR_ispResolutionCmdforAOI() 함수에서 설정한 AOI영역의 현재 시작점을 알고, 임의의 점으로 이동시킬 수 있다.

구분	내용	
Syntax	<pre>Int HVR_ispMoveStartPosOnAOI(int ProductNo, int DeviceNo, int *currX, int *currY, int MoveX, int MoveY);</pre>	
Parameter	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
	currX	현재의 세로 시작점
	currY	현재의 가로시작점
	moveX	이동할 세로 시작점, 미사용시 -1
	moveY	이동할 가로 시작점, 미사용시 -1
Return Value	1이면 동작, 0이면 동작하지 않음을 의미	
Sample	<pre>BOOL re; Int prevX, prevY; Re = HVR_ispMoveStartPosOnAOI(HVR2300R, 0, &prevX, &prevY, -1, -1); // X 축 50pixel 이동, Y 축 50pixel 이동 prevx += 50; prevY += 50; re = HVR_ispMoveStartPosOnAOI(HVR2300R, 0,NULL,NULL, &prevX, &prevY); If(!re) return FALSE;</pre>	

6. API 설명

6.10 HVR_ispVBlankAdjustment

카메라 영상의 Frame간 간격을 조절한다.

#사용 목적

- [1] 카메라 자체 프레임 속도를 조절하기 위해 사용한다.
- [2] 개발자가 구현한 데이터 출력 및 디스플레이 알고리즘에 의해 프레임저하 현상이 나타날 경우, 개선책으로 사용할 수 있다.
- [3] Exposure Time 조절로 인한 프레임 저하현상에 대응할 수 있다.

단, 본 내용은 출력해상도별 간격이 일정하지 않으므로 일정 Range 내에서만 유효하다.

구분	내용	
Syntax	<pre>Int HVR_ispVBlankAdjustment(int blank, int ProductNo, int DeviceNo);</pre>	
Parameter	blank	사용 제품의 모델별 range값 입력
	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않음을 의미	
Sample	<pre>BOOL re; Int min,max; Int iblack; // Range : min – max HVR_ispVBlankAdjustmentRange(&min, &max, &iblack, HVR2300R, 0); iblack += 10; Re = HVR_ispVBlankAdjustment(iblack, HVR2300R, 0); If(!re) return FALSE;</pre>	

6. API 설명

6.11 HVR_ispVBlankAdjustmentRange

HVR_ispVBlankAdjustment()를 위한 카메라 영상의 Frame간 간격 값의 Range를 얻는다.

구분	내용	
Syntax	<pre>Int HVR_ispVBlankAdjustmentRange(int *pMin, int *pMax, int *pValue, int ProductNo, int DeviceNo);</pre>	
Parameter	pMin	V-Blank 의 최소값
	pMax	V-Blank의 최대값
	pValue	V-Blank의 현재값
	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않음을 의미	
Sample	<pre>BOOL re; Int min,max; Int iblack; // Range : min – max HVR_ispVBlankAdjustmentRange(&min, &max, &iblack, HVR2300R, 0); iblack += 10; Re = HVR_ispVBlankAdjustment(iblack, HVR2300R, 0); If(!re) return FALSE;</pre>	

6. API 설명

6.12 HVR_espGetExposureRange

Exposure Time(빛을 받아들이는 속도)의 범위(min~max)를 구하고 입력값을 얻는다.

구분	내용	
Syntax	<pre>Int HVR_espGetExposureRange(int *pMin, int *pMax, int *pValue, int ProductNo, int DeviceNo);</pre>	
Parameter	pMin	Exposure Time 의 최소값
	pMax	Exposure Time의 최대값
	pValue	Exposure Time의 현재값
	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않음을 의미	
Sample	<pre>BOOL re; Int min,max; Int icurr // Range : min – max Re = HVR_espGetExposureRange(&min, &max, &icurr, HVR2300R, 0); If(!re) return FALSE;</pre>	

6. API 설명

6.13 HVR_ispSetExposureTime

Exposure Time(빛을 받아들이는 속도)를 Manual로 조절한다.

구분	내용	
Syntax	<pre>Int HVR_ispSetExposureTime(int *pValue, int ProductNo, int DeviceNo);</pre>	
Parameter	pValue	원하는 Exposure Time값
	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않음을 의미	
Sample	<pre>BOOL re; Int min,max; Int icurr // Range : min – max Re = HVR_ispGetExposureRange(&min, &max, &icurr, HVR2300R, 0); If(!re) return FALSE; Icurr += 10; Re = HVR_ispSetExposureTime(icurr, HVR2300R, 0);</pre>	

6. API 설명

6.14 HVR_ispGetAWBGainControlRange

R/G/B 색상 각각의 Gain 값과 범위를 얻는다.

구분	내용	
Syntax	<pre>Int HVR_ispGetAWBGainControlRange(int iMode, int *pMin, int *pMax, int *pValue, int ProductNo, int DeviceNo);</pre>	
Parameter	iMode	얻어오고 싶은 색상 GAIN_RED, GAIN_GREEN, GAIN_BLUE
	pMin	해당 색상의 최소값
	pMax	해당 색상의 최대값
	pValue	현재의 Gain값
	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않음을 의미	
Sample	<pre>BOOL re; Int min,max; Int icurr // Range : min - max Re = HVR_ispGetAWBGainControlRange(GAIN_RED, &min, &max, &icurr, HVR2300R, 0); If(!re) return FALSE; Icurr += 10; Re = HVR_ispSetAWBGainControl(GAIN_RED,icurr, HVR2300R, 0);</pre>	

6. API 설명

6.15 HVR_ispSetAWBGainControl

R/G/B 색상 각각의 Gain 값을 Manual로 조절한다.

구분	내용	
Syntax	<pre>Int HVR_ispSetAWBGainControl(int iMode, int iGain, int ProductNo, int DeviceNo);</pre>	
Parameter	iMode	설정하고 싶은 색상 GAIN_RED, GAIN_GREEN, GAIN_BLUE
	iGain	원하는 Gain값
	ProductNo	사용 제품의 모델명을 입력한다.
	DeviceNo	카메라의 순번을 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않음을 의미	
Sample	<pre>BOOL re; Int min,max; Int icurr // Range : min – max Re = HVR_ispGetAWBGainControlRange(GAIN_RED, &min, &max, &icurr, HVR2300R, 0); If(!re) return FALSE; icurr += 10; Re = HVR_ispSetAWBGainControl(GAIN_RED,icurr, HVR2300R, 0);</pre>	

6. API 설명

6.16 HVR_ispRawToBmp

카메라에서 출력된 RAW 포맷의 영상데이터를 Bitmap포맷으로 변경해 주는 기능을 한다.

구분	내용	
Syntax	<pre>Int HVR_ispRawToBmp(int iOutMode, int width, int height, BYTE *pRawBuffer, BYTE *pBMPBuffer, int ProductNo);</pre>	
Parameter	iOutMode	출력영상의 data format을 설정한다. BGGR, RGGB, GBRG, GRBG, BlackWhite
	width	출력영상의 가로 사이즈
	Height	출력영상의 세로 사이즈
	pRawBuffer	카메라에서 받은 영상 데이터의 버퍼 포인터
	pBMPBuffer	변경이미지를 저장할 버퍼포인터
	ProductNo	사용 제품의 모델명을 입력한다.
Return Value	1이면 정상동작, 0이면 동작하지 않음을 의미	
Sample	<pre>Int reData; BYTE *pRAWBuf, pBMPBuf; pRAWBuf = new BYTE[640*480], pBMPBuf = new BYTE[640*480*3]; reData = HVR_camGetImageData(640*480, pRAWBuf, 0); If(reData == USB_DATA_PASS) { HVR_ispRAWToBmp(GRBG, 640*480, pRAWBuf, HVR2300R); } Delete pRAWBuf; Delete pBMPBuf;</pre>	