



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

HyVISION System 標準文書

HVS API Programming(2.0Ver.)

Product Classification :HVS API(2.0Ver.)
Product Name: HVS New Platform API Document

REVISION CODE; Version 1.0

Last update; August 7, 2014

	Charge	Investigation	Investigation	Admission
SIGN				
DATE				
NAME				

Specification and information in this document are subject to change without notice and do not represent a commitment on the part of HyVISION System. HyVISION System reserves the right to make changes to improve functioning.

This document contains trade secrets and confidential information, and is furnished pursuant to a license to the user from HyVISION System. Use or reproduction of this document is restricted under the terms of the license.

© Copyright 2006-2014 HyVISION System Inc.

Printed in Korea

For further information please contact:

HyVISION System Inc.

Head office: SuntechCity II Suite 707, 307-2, Sangdaewon1-Dong, Jungwon-Ku, Sunnam-Si,

Gyeonggi-Do, 462-120, Korea

PHONE: +82-31-735-1573

FAX: +82-31-735-1574

Web site: <http://www.hyvision.co.kr>

E-mail (rep): hyvision@hyvision.co.kr

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	2 of 73

목차

◆HVS New Platform API.....	5
SW Base APIs	5
HW Base APIs	28
PlatformMemory APIs	43
UserProtocol APIs.....	46
#Appendix A : Structure.....	56
#Appendix B : HVS Data Define Constant.....	62
#Appendix C : Sample Project	68

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	3 of 73

Document Revision History

Version	Issue Date	Comments
1.0.0	7, August, 2014	Initial Release

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	4 of 73

◆HVS New Platform API

SW Base APIs

OverView

BOOL	SetHVS_Data(void *pData)
void	GetTestInfo(TestInfo *pData)
HWND	GetPlatformMainHWND(HWND hWnd)
BOOL	GetImage(BYTE *pDataBuf, BYTE *pDataBuf_CRC, int nWidth, int nHeigh,int nDataFormat, int nDummyFrame, int nAvgFrame, BOOL *bResult_CRC, BOOL *bResult_ECC)
BOOL	GetImage_General(BYTE *pDataBuf, int nWidth, int nHeigh,int nDataFormat, int nDummyFrame, int nAvgFrame)
void	Input_cvImage(IplImage *imgInput)
void	Input_Image_Buffer(BYTE *pBuffer,UINT nWidth, UINT nHeight, BOOL bIsPreview)
void	Output_Image()
BOOL	OpenImage(LPSTR sFileName, UINT nWidth, UINT nHeight, BYTE* pDstBuffer)
void	MakeRGBPattern(BYTE *destBuffer, unsigned int nWidth, unsigned int nHeight)
void	Convert_Image(int nDataMode, int nRGBMode, LPBYTE srcBuffer, LPBYTE destBuffer, int nWidth, int nHeight)
void	Convert_Image_Ex(int nDataMode, int nRGBMode, LPBYTE srcBuffer, LPBYTE destBuffer, int nWidth, int nHeight, int nRealX, int nRealY, int nOffsetX, int nOffsetY)
void	ListOut(CString str)
void	ListOutInx(CString str, ...)
void	Delay(UINT nDelay)
void	SaveImage(char* sPath, char *sFileName, char *sFormat, BYTE* pBuffer, int nWidth, int nHeight);
BOOL	GetRepeatTest_IsStop()
void	SetRepeatTest_IsStop(BOOL bIsStop);
void	SetTestBuffer(void *pData);

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	5 of 73

void*	GetTestBuffer();
void	SetRevPointer(UINT nAddr, void *p);
void*	GetRevPointer(UINT nAddr);
void	SetDeviceMemory(void *pData);
void*	GetDeviceMemory();
int	ReadINI_Int(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, INT nDefault);
float	ReadINI_Float(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, float fDefault);
double	ReadINI_Double(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, double dDefault);
BOOL	ReadINI_String(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, LPCSTR lpDefault, LPTSTR lpReturn, DWORD nSize);
BOOL	WriteINI_Int(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, INT nData);
BOOL	WriteINI_Float(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, float fdata);
BOOL	WriteINI_Double(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, double ddata);
BOOL	WriteINI_String(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, LPCSTR str);
void	SetDlgItemFloat(HWND hDlg, int nIDItem, float fSetValue, int npoint)
float	GetDlgItemFloat(HWND hDlg, int nIDItem)
void	SetAvisLightReg(int nCh)
void	SetYoothtechLightReg(int nCh)
void	SetFocusLightReg(int nLevel)
void	SetImageLightReg(int nLevel)

SetHVS_Data

구조체 멤버 포인터(Config*)의 주소를 Set 한다. DLL 호출시 호출하는 함수
(플랫폼 실행 후 공유 변수 사용을 위해 필요)

BOOL **SetHVS_Data(void *pData);**

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	6 of 73

PARAMETERS

pData[in]: void형 Pointer

RETURN VALUE

pData가 NULL인 경우 RET_NG(1) 리턴

Parameter Set 성공시 RET_OK(0) 리턴

EXAMPLE

See #Appendix C : Sample Project

GetTestInfo

Test dll에 필요한 정보를 얻어오는 함수. TestInfo구조체에 정보를 대입.

void GetTestInfo(TestInfo *pData);

PARAMETERS

pData[in, out]: TestInfo 구조체([Appendix A : Structure](#) 참고)

RETURN VALUE

None

EXAMPLE

See #Appendix C : Sample Project

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	7 of 73

GetPlatformMainHWND

해당 dll의 Parent Window Handle을 반환한다.

HWND GetPlatformMainHWND(HWND hWnd);

PARAMETERS

HWND[in]: Input Window Handle

RETURN VALUE

Input Window Handle의 Parent Window Handle

EXAMPLE**GetImage**

이미지 Capture 함수. Capture 후 버퍼에 데이터 저장(CRC Packet 검사 추가 함수)

GetImage (BYTE *pDataBuf, BYTE *pDataBuf_CRC, int nWidth, int nHeigh,int nDataFormat, int nDummyFrame, int nAvgFrame, BOOL *bResult_CRC, BOOL *bResult_ECC);

PARAMETERS

pDataBuf[in, out]: Capture 후 Average Image Buffer

pDataBuf_CRC[in, out]: CRC Packet 검사용 Buffer(각 열의 추가 Packet 4Bit를 검사함)

nWidth[in]: Image Width

nHeight[in]: Image Height

nDummyFrame[in]: Dummy Frame

nDataFormat[in]: Sensor Data Format

if nDataMode == 1 8Bit Bayer Mode

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	8 of 73


```

if nDataMode == 2  10Bit Bayer Mode
if nDataMode == 3  12Bit Bayer Mode
else YUV Mode

```

nAvgFrame[in]: Capture Count(Average)
 bResult_CRC[in, out]: CRC Result(1: FAIL, 0: PASS)
 bResult_ECC[in, out]: Always 0(PASS), 현재 지원하지 않는 기능

RETURN VALUE

Capture 성공시 TRUE(1), 실패시 FALSE(0) Return.

EXAMPLE**GetImage_General**

이미지 Capture 함수. Capture 후 버퍼에 데이터 저장(일반 Capture 기능)

GetImage_General (BYTE *pDataBuf, int nWidth, int nHeight, int nDataFormat, int nDummyFrame, int nAvgFrame);

PARAMETERS

pDataBuf[in, out]: Capture 후 Average Image Buffer
 nWidth[in]: Image Width
 nHeight[in]: Image Height
 nDummyFrame[in]: Dummy Frame
 nDataFormat[in]: Sensor Data Format
 if nDataMode == 1 8Bit Bayer Mode
 if nDataMode == 2 10Bit Bayer Mode

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	9 of 73

```
        if nDataMode == 3    12Bit Bayer Mode
        else YUV Mode
nAvgFrame[in]: Capture Count(Average)
```

RETURN VALUE

Capture 성공시 TRUE(1), 실패시 FALSE(0) Return.

EXAMPLE**Input_cvImage**

출력 영상 Set함수. 주로 원본 영상에 Text나 도형 개체를 입힌 영상을 출력할 때 사용.

Remarks

OpenCV 함수 사용

```
void Input_cvImage(IplImage *imgInput);
```

PARAMETERS

imgInput: OpenCV IplImage 구조체 포인터

RETURN VALUE

없음

EXAMPLE

```
CvFont Font;
```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	10 of 73

```

cvInitFont(&Font, CV_FONT_HERSHEY_SIMPLEX, 1.0f, 1.0f);
char szText[50] = {0, };
sprintf(szText, "%s", "Capture Image Draw");
cvPutText(m_TestInfo.imgIntP, szText, cvPoint(50, 100), &Font, CV_RGB(255, 255, 0));

```

```

SYSTEMTIME st;
GetLocalTime(&st);

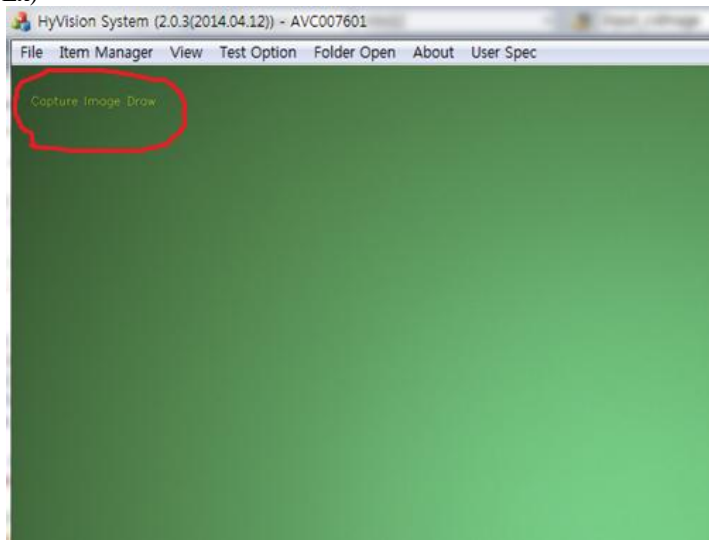
```

```

Input_cvImage(m_TestInfo.imgIntP);

```

Ex)



Input_Image_Buffer

Draw를 위한 Buffer Set함수. unsigned char(BYTE)형 데이터를 인자로 넘겨주면 Platform의 내부 IplImage 구조체 Type에 데이터를 복사해둠. (주로 원본영상을 Set함)

참고사항: Platform 실행시 출력 영상 2가지의 Size에 대한 메모리를 생성해 둠. Normal한 Size는 해당 모델.ini에 [SENSOR_SET] HVS_SENSOR_SIZE_X,

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	11 of 73

HVS_SENSOR_SIZE_Y이며 변형된 size는 [REGISTER_PREVIEW]의 HVS_SENSOR_SIZE_X, HVS_SENSOR_SIZE_Y임

void Input_Image_Buffer(BYTE *pBuffer,UINT nWidth, UINT nHeight, BOOL bIsPreview)

Remarks

다음은 Normal한 Size 및 또 다른 사이즈 생성시의 Setting 예시이다. (Ref 폴더의 해당모델명.ini File로부터 읽어온다.)

[SENSOR_SET]

HVS_SENSOR_VCLK=0

HVS_SENSOR_RESET=1

HVS_SENSOR_ENB=1

HVS_SENSOR_VSYNC=0

HVS_SENSOR_SLAVEADDR=0x10 //0x36 //6C

HVS_SENSOR_SIZE_X=3280

HVS_SENSOR_SIZE_Y=2464

HVS_SENSOR_DATA_FORMAT=2 //0:YUV 1:8bit Bayer 2: 10bit Bayer

HVS_SENSOR_OUT_MODE=1

HVS_SENSOR_MIPI_MODE=7 //0:1L 1:2L 3:3L 7:4L

HVS_SENSOR_I2C_MODE=1 //0:8_8 1:8_16 2:16_8 3:16_16

HVS_SENSOR_2STEPLAY=0

HVS_SENSOR_2STEPLAY_PLAYTIME=500

HVS_SENSOR_SleepTime=0

[REGISTER_PREVIEW]

HVS_SENSOR_SIZE_X = 1280

HVS_SENSOR_SIZE_Y = 960

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	12 of 73

PARAMETERS

pBuffer[in]: input Buffer

nWidth[in]: image Width

nHeight[in]: image Height

bIsPreview[in]: Sensor Size로 출력(TRUE일 경우 [SENSOR_SET] , FALSE일 경우 [REGISTER_PREVIEW])

RETURN VALUE

None

EXAMPLE**Output_Image**

이미지 출력 함수(Platform Main Window Area에 Draw)

void Output_Image();

PARAMETERS

None

RETURN VALUE

None

EXAMPLE**OpenImage**

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	13 of 73

현재 지원하지 않은 함수(Image Load 함수임)

BOOL OpenImage(LPSTR sFileName, UINT nWidth, UINT nHeight, BYTE* pDstBuffer);

PARAMETERS

sFileName[in]: FileName.

nWidth[in]: Image Width

nHeight[in]: Image Height

pDstBuffer[out]: Open한 파일을 저장할 버퍼

RETURN VALUE

Open에 성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE

MakeRGBPattern

출력 확인용 함수(R, G, B 이미지 화면에 출력). Image Size만큼 Buffer에 R, G, B를 채운다.

Ex)



Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	14 of 73

```
void MakeRGBPattern(BYTE *destBuffer, unsigned int nWidth, unsigned int nHeight);
```

PARAMETERS

destBuffer[in]: 출력용 버퍼(r, g, b를 표현한다.)

nWidth[in]: Image Width

nHeight[in]: Image Height

RETURN VALUE

None

EXAMPLE**Convert_Image**

8bit or 10bit or 12 bit Bayer Raw Image 버퍼를 저장하거나 화면에 출력할 수

있도록 8비트의 비트맵 형의 이미지 데이터로 변환하는 함수임(현재 V4P, V5U 둘 다 출력 가능)

Remarks

1. 8bit일 경우 Raw -> RGB 변환(Color Mode별 Interpolation 수행)
2. 10bit일 경우 10Bit Raw -> 8bit raw -> RGB 변환(Color Mode별 Interpolation 수행)
3. 12bit일 경우 12Bit Raw -> 8bit raw -> RGB 변환(Color Mode별 Interpolation 수행)

```
void Convert_Image(int nDataMode, int nRGBMode, LPBYTE srcBuffer, LPBYTE destBuffer, int nWidth, int nHeight);
```

PARAMETERS

nDataMode[in]:

if nDataMode == 1 8Bit Bayer Mode

if nDataMode == 2 10Bit Bayer Mode

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	15 of 73

```

    if nDataMode == 3  12Bit Bayer Mode
    else YUV Mode

```

nRGBMode: Color Order(1: BGGR 2: RGG 3: GRGB 4: GRBG 5: Black White)

srcBuffer[in]: 입력 버퍼

destBuffer[out]: 출력 버퍼

nWidth[in]: Image Width

nHeight[in]: Image Height

RETURN VALUE

None

EXAMPLE

Convert_Image_Ex

R3 사용하며 12Bit Bayer Format을 사용하는 경우 이 함수를 사용한다.

Image Width * 1.5가 4로 나누어 질 수 없는 경우 size 2를 더한다.

```

void Convert_Image_Ex(int nDataMode, int nRGBMode, LPBYTE srcBuffer, LPBYTE destBuffer, int nWidth, int nHeight, int nRealX, int nRealY,
int nOffsetX, int nOffsetY);

```

PARAMETERS

nDataMode[in]:

```

    if nDataMode == 1  8Bit Bayer Mode

```

```

    if nDataMode == 2  10Bit Bayer Mode

```

```

    if nDataMode == 3  12Bit Bayer Mode

```

```

    else YUV Mode

```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	16 of 73

nRGBMode: Color Order(1: BGGR 2: RGG 3: GRGB 4: GRBG 5: Black White)

srcBuffer[in]: 입력 버퍼

destBuffer[out]: 출력 버퍼

nWidth[in]: Image Width

nHeight[in]: Image Height

RETURN VALUE

None

EXAMPLE

ListOut

Platform ListOut Dialog에 String 출력(직접 형식 지정 불가능)
(주로 개발자들이 Program Debugging 위해 사용함.)

void ListOut(CString str);

PARAMETERS

CString(MFC Class)

RETURN VALUE

없음

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	17 of 73

ListOutInx

Platform ListOut Dialog에 String 출력(형식 지정이 가능)
(주로 개발자들이 Program Debugging 위해 사용함.)

void ListOutInx(CString str, ...);

PARAMETERS

CString(MFC Class)[in]
(형식 지정자)

RETURN VALUE

None

EXAMPLE

Delay

Delay(ms 단위). 10sec 이상은 10sec으로 예외처리 되어 있음.

void Delay(UINT nDelay);

PARAMETERS

nDelay[in]: ms 단위의 Delay Count

RETURN VALUE

None

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	18 of 73

EXAMPLE**SaveImage**

이미지 저장 함수. 내부적으로 OpenCV 함수를 사용함. cvSaveImage 함수(관련 사항은 OpenCV 관련 문서 별도 참고)

BOOL SaveImageFile(char *sPath, char *sFileName, char *sFormat, BYTE* pBuffer, int nWidth, int nHeight);

PARAMETERS

sPath[in]: 저장할 이미지 경로

sFileName[in]: 저장할 이미지 이름

sFormat[in]: 저장할 이미지 형식(OpenCV 관련 Library 참고)

pBuffer[in]: 저장할 버퍼

nWidth[in]: Image Width

nHeight[in]: Image Height

RETURN VALUE

성공시 TRUE

실패시 FALSE

GetRepeatTest_IsStop

User Stop 여부 확인 함수(Ex: User가 장비 Stop Button 눌렀을 경우 내부적으로 사용하는 변수)

BOOL GetRepeatTest_IsStop();

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	19 of 73

PARAMETERS

None

RETURN VALUE

TRUE or FALSE(TEST중 항상 FALSE, User Stop시 TRUE)

EXAMPLE**SetRepeatTest_IsStop**

Test Stop 여부를 지정하는 함수

void SetRepeatTest_IsStop(BOOL bIsStop);**PARAMETERS**

bIsStop[in]: 일반적인 Test시 항상 FALSE로 지정

RETURN VALUE

None

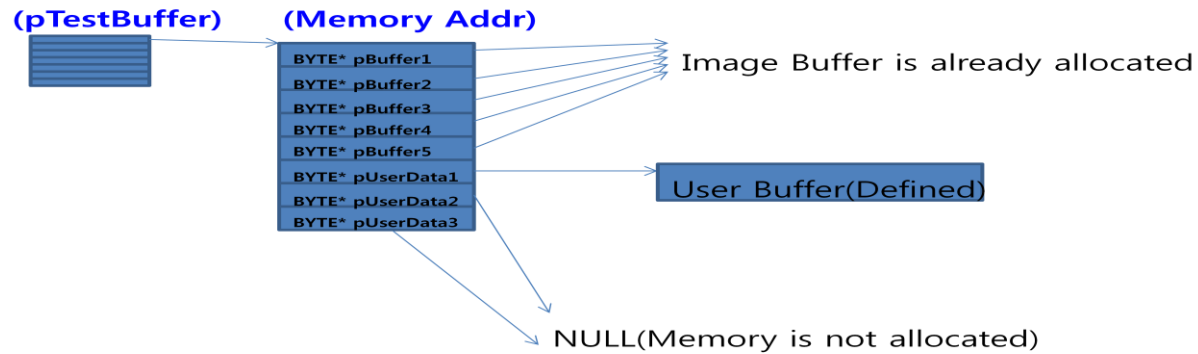
EXAMPLE**SetTestBuffer**

User가 Heap 영역에 할당한 메모리의 시작 주소를 지정하는 함수

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	20 of 73

Remarks

다음은 SetTestBuffer가 사용 방식에 대한 예시 구조도이다. (내부적으로 SetTestBuffer 함수를 사용하여 Memory의 시작 번지를 기억하며, GetTestBuffer함수를 사용하여 주소값으로 메모리에 접근할 수 있다.)



void SetTestBuffer(void *pData)

PARAMETERS

void* pData[in, out]: void* 형이므로 어떤 데이터형도 인자가 될 수 있음

RETURN VALUE

None

EXAMPLE

//Dll 공용 buffer를 생성/반환 한다.

```

pTestBuffer GetImageBuffer (UINT nWidth, UINT nHeight)
{
    pTestBuffer pData;
    //Create test buffer memory
    if(GetTestBuffer() == NULL)
  
```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	21 of 73

```
{
    ListLog("Testdll: create buffer (%d, %d)", nWidth, nHeight);
    ListLog("Debug: Create buffer");

    pData = new TestBuffer;

    pData->pBuffer1 = new BYTE[nWidth * nHeight * 3];
    pData->pBuffer2 = new BYTE[nWidth * nHeight * 3];
    pData->pBuffer3 = new BYTE[nWidth * nHeight * 3];
    pData->pBuffer4 = new BYTE[nWidth * nHeight * 3];
    pData->pBuffer5 = new BYTE[nWidth * nHeight * 3];

    memset(pData->pBuffer1, 0x00, sizeof(BYTE)*(nWidth * nHeight * 3));
    memset(pData->pBuffer2, 0x00, sizeof(BYTE)*(nWidth * nHeight * 3));
    memset(pData->pBuffer3, 0x00, sizeof(BYTE)*(nWidth * nHeight * 3));
    memset(pData->pBuffer4, 0x00, sizeof(BYTE)*(nWidth * nHeight * 3));
    memset(pData->pBuffer5, 0x00, sizeof(BYTE)*(nWidth * nHeight * 3));

    pData->pUserBuffer0 = NULL;
    pData->pUserBuffer1 = NULL;
    pData->pUserBuffer2 = NULL;

    SetTestBuffer(pData);
}
else
{
    pData = (pTestBuffer)GetTestBuffer();
}
return pData;
}
```

GetTestBuffer

SetTestBuffer 함수를 통해 Set한 버퍼의 시작 주소를 Get하는 함수(User가 Heap 영역에 할당한 메모리에 접근하기 위해 필요한 함수임)

Remarks

SetTestBuffer Remark 참고

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	22 of 73

void* **GetTestBuffer()**

PARAMETERS

None

RETURN VALUE

void*[in, out]: void* 형이므로 어떤 데이터형도 리턴할 수 있음.

EXAMPLE

SetRevPointer

여분의 메모리 버퍼 포인터 변수에 할당한 버퍼의 시작 주소를 Set하는 함수(0 ~ 9 의 Reserved 변수 사용 가능)

void **SetRevPointer(UINT nAddr, void *p)**

PARAMETERS

UINT nAddr[in]: 0 ~ 9까지의 정수

void* pData[in, out]: void* 형이므로 어떤 데이터형도 인자가 될 수 있음

RETURN VALUE

None

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	23 of 73

GetRevPointer

SetRevPointer로 지정한 변수를 이용해 생성된 메모리의 시작 주소를 얻어오는 함수(0 ~ 9 의 Reserved 변수 사용 가능)

void* GetRevPointer(UINT nAddr)

PARAMETERS

nAddr[in]: 0 ~ 9까지의 정수

RETURN VALUE

void*[in, out]: 해당 Address의 시작주소(0 ~9까지 사용 가능)

EXAMPLE**ReadINI_[Int, Float, Double, String] : Override 함수임**

해당 경로의 INI 파일에서 [KEY값] 에 해당하는 Value값을 Return

int ReadINI_VARIANT(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, VARIANT vDefault)

PARAMETER

LPCSTR lpFile[in]: 경로명

LPCTSTR lpAppName[in]: 키 값

LPCTSTR lpKey[in]: Value

VARIANT vDefault; Default 값(int, float, double, string)

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	24 of 73

RETURN

(int, float, double, BOOL)

EXAMPLE**WriteINI_[Int, Float, Double, String] : Override 함수임**

해당 경로의 INI 파일에서 [KEY값] 에 해당하는 Value값을 Set

BOOL WriteINI_VARIANT(LPCSTR lpFile, LPCTSTR lpAppName, LPCTSTR lpKey, VARIANT Data)

PARAMETER

LPCSTR lpFile[in]: 경로명

LPCTSTR lpAppName[in]: 키 값

LPCTSTR lpKey[in]: Value

VARIANT Data; Write할 Data(int, float, double, string)

RETURN

BOOL: 성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE**SetAvisLightReg**

Avis 광원 LED Level을 Registry에 기록한다. (경로: HKEY_CURRENT_USER > Avis > Lamp)

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	25 of 73

```
void SetAvisLightReg(int nLevel);
```

PARAMETER

nLevel[in]: 정수형 led value

RETURN

None

EXAMPLE**SetYoothtechLightReg**

Yoothtech 광원 LED Level을 Registry에 기록한다. (경로: HKEY_CURRENT_USER > YoothTech > Lamp)

```
void SetYoothtechLightReg(int nLevel);
```

PARAMETER

nLevel[in]: 정수형 led value

RETURN

None

EXAMPLE**SetFocusLightReg**

Focus 광원 LED Level을 Registry에 기록한다. (경로: HKEY_CURRENT_USER > Focus > Lamp)

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	26 of 73

```
void SetFocusLightReg(int nLevel);
```

PARAMETER

nLevel[in]: 정수형 led value

RETURN

None

EXAMPLE**SetImageLightReg**

Image 광원 LED Level을 Registry에 기록한다. (경로: HKEY_CURRENT_USER > Image > Lamp)

```
void SetImageLightReg(int nLevel);
```

PARAMETER

nLevel[in]: 정수형 led value

RETURN

None

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	27 of 73

HW Base APIs

OverView

void	SetConfigMemory(void* pData)
BOOL	HW_INIT(void *pData)
int	GetHVS_DeviceFailCode()
BOOL	HW_PowerUpSequence(UINT nHWMMode = 0)
BOOL	HW_PowerDownSequence(UINT nHWMMode = 0)
BOOL	HW_RegisterSet(LPSTR sCommand, UINT nHWMMode = 0)
void	OnStandbyOffset(int nIndex)
BOOL	HW_IIC_Write(DWORD wAddr, DWORD wData)
BOOL	HW_IIC_Write_SlaveAddr(DWORD wAddr, DWORD wData, WORD wSlaveAddr)
BOOL	HW_IIC_Write_SelMode(INT mode, DWORD wData)
BOOL	HW_IIC_Read(DWORD wAddr, DWORD *wData)
BOOL	HW_IIC_Read_SlaveAddr(DWORD wAddr, DWORD *wData, WORD wSlaveAddr)
BOOL	HW_IIC_Read_SelMode(INT mode, DWORD wAddr, DWORD *wData)
BOOL	HW_IIC_Read_BULK(INT read_cnt, CHAR *addr_buf, CHAR *read_buf)
BOOL	HW_IIC_Write_BULK(CHAR *write_buf)
BOOL	HW_StartDevice()
int	GetID_USBV4()
int	GetIndex_USBV4(LPSTR UsbBoardID_V4)
void*	OnHWDlg(CWnd *pParentWnd)
BOOL	HW_RegisterSet_R3(LPSTR lpcommand, UINT nHWMMode = 0, UINT nGpioLowHigh = 0)

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	28 of 73

SetConfigMemory(void* pData)

프로그램 시작시 1회 이 함수를 사용함. (Test Item 개발자는 이 함수를 절대로 사용하지 말 것)

Config 구조체(Structure 표 참고)의 메모리를 Heap 영역에 할당시켜주는 함수. 프로그램 종료 전까지 이 메모리는 없어지지 않음.

void SetConfigMemory(void* pData)**PARAMETERS**

*pData[in, out]: 어떤 데이터형(구조체 포함)이라도 인자로 넘길 수 있음(void형 pointer)

RETURN

None

EXAMPLE**HW_INIT**

프로그램 시작시 1회 이 함수를 사용함. (Test Item 개발자는 이 함수를 절대로 사용하지 말 것)

장치를 구동시키기 위한 초기 동작 정의 함수 (V4 계열, V5 계열의 Device Open 함수)

BOOL HW_INIT(void *pData);**PARAMETERS**

pData[in, out]

RETURN

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	29 of 73

초기화 성공시 TRUE RETURN, 실패시 FALSE RETURN

EXAMPLE

GetHVS_DeviceFailCode

HW_INIT시 발생한 FAIL Code를 얻어오는 함수.(Default 0)

HVS Platform 시작시 특정 메모리 영역에 데이터를 동적 생성하여 저장. 이 내부 메모리 변수 중 Device Fail Code를 가져온다.

```
int GetHVS_DeviceFailCode();
```

PARAMETERS

None

RETURN VALUE

0: PASS

1: Board Type 정의되지 않은 장치 Index 입력시 FAIL(Board Type 정의시 오류. 현재 1: V4I, 2: V4P, 3:V5U, 4:V5U_R2)

2: 정상적으로 장치를 Open하지 못했을 경우 FAIL

3: V4 계열 관련 dll 로드 FAIL

4: V5 관련 dll 로드 FAIL

EXAMPLE

HW_PowerUpSequence

Board Power를 ON 한다.

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	30 of 73

Sensor ini에 있는 [HVS_HW] HW_BOARD_TYPE 참고(1:V3, 2:V4, 3:V5)

BOOL HW_PowerUpSequence();

PARAMETERS

None

RETURN VALUE

파워 On 성공시 TRUE RETURN, 실패시 FALSE RETURN

EXAMPLE

HW_PowerDownSequence

Board Power를 OFF 한다.

Sensor ini에 있는 [HVS_HW] HW_BOARD_TYPE 참고(1:V3, 2:V4, 3:V5)

BOOL HW_PowerDownSequence();

PARAMETERS

None

RETURN VALUE

파워 OFF 성공시 TRUE RETURN, 실패시 FALSE RETURN

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	31 of 73

OnStandbyOffset

Default로 Socket 에 흐르는 전류를 측정한다.

Remarks

Standby Current Test는 Sleep Mode 상태에서 모듈 핀에 흐르는 전류를 측정한다. 측정시 소켓과 모듈에 같이 흐르는 전류를 측정할 수 밖에 없기 때문에 Module이 없는 상태에서 Current를 측정후 File에 기록. Module+Socket 전류 측정 – Socket Default 전류 과정을 거친다.

BOOL OnStandbyOffset(int nIndex);

PARAMETERS

nIndex[in]: 각 USB Index

RETURN VALUE

Offset 성공시 TRUE, 실패시 FALSE

EXAMPLE**HW_RegisterSet**

Sensor Register setting의 Address, Value를 Set

RefModel_Register.ini에서 [] 해당하는 Address에 해당하는 Value를 Set한다.

Remarks

HVS는 Module의 기억소자(Register)를 제어할 수 있으며 특정 명령어를 이용하여 기능을 동작시킬 수 있도록 정의되어 있다. 다음은 명령어의

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	32 of 73

예시이다. (대소문자 상관없이 사용 가능)

참고1: 모든 Register Key에 대한 끝은 항상 [END]로 정의한다.

참고2: I2C Mode를 고려해야 한다. 현재 8Bit to 8Bit, 8Bit to 16Bit, 16Bit To 8Bit, 16Bit To 16Bit이 정의되어 있다.

참고3: Address에 대한 Data를 I2C 통신 규약을 이용해 Control 가능하도록 프로그래밍 되어 있다. (I2C Write 기능)
EX)

[REGISTER]

```
ENB 1          //HW Enable On/Off
RESET 1        //HW Reset On/Off
MCLK 1         //MCLK On/Off 정의가 가능하다.
I2CM 2         //I2C Mode를 변경할 수 있다.
SLAVE 0x0c     //Slave Address 전환 가능
SLEEP 10       //Delay 기능(단위 ms)
GPIO_1 1       //Global Port I/O 동작을 정의할 수 있다.
0x30EB 0x05    // Register
0x30EB 0x0C    // Register
0x300A 0xFF    // Register
0x300B 0xFF    // Register
0x30EB 0x05    // Register
0x30EB 0x09    // Register
```

[END]

BOOL HW_RegisterSet(LPSTR sCommand);

PARAMETERS

sCommand[in]: 사용자 정의 Key

EX) [AE_SETTING] -> Command의 인자로 들어간다.

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	33 of 73

0x3601 0x11

[END]

RETURN VALUE

RegisterSet 성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE

HW_IIC_Write

해당 Address에 Value값을 Write하는 함수 [SENSOR_SET]의 HVS_SENSOR_I2CMODE에 설정한 Mode로 작동한다.

Remarks

HVS HW Base에는 I2C 통신 Protocol이 정의되어 있다. 따라서 기억소자(Register)를 Control할 수 있다.

BOOL HW_IIC_Write(DWORD wAddr, DWORD wData);

PARAMETERS

DWORD wAddr[in] : 센서의 Address 값

DWORD wData[in] : 해당 Address Value 값

RETURN VALUE

성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	34 of 73

HW_IIC_Write_SelMode

IIC_Write_SelMode와 같은 기능을 하는 함수. (Mode를 바꿔서 I2 Write를 할 수 있다.)

Remarks

HW_IIC_Write는 실제로 특정 파일(ini) 기록된 I2C Mode를 읽어와 Setting 해주지만 본 함수는 ini에서 값을 읽어와서 Mode를 Setting해 주는 것이 아니라 개발자가 i2c Mode를 인자로 넣을 수 있는 함수이다.

BOOL HW_IIC_Write(INT mode, DWORD wAddr, DWORD wData);

PARAMETERS

INT mode[in]: I2C Mode(0:I2C_A8_D8, 1:I2C_A8_D16, 2: I2C_A16_D8, 3:I2C_A16_D16)

DWORD wAddr[in] : 센서의 Address 값

DWORD wData[in] : 해당 Address Value 값

RETURN VALUE

성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE**HW_IIC_Read**

해당 Address에 Value값을 Read하는 함수 [SENSOR_SET]의 HVS_SENSOR_I2CMODE에 설정한 Mode로 작동한다.

Remarks

HVS HW Base에는 I2C 통신 Protocol이 정의되어 있다. 따라서 기억소자(Register)에 Write된 값을 Read할 수 있다.

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	35 of 73

BOOL HW_IIC_Read(DWORD wAddr, DWORD *wData);

PARAMETERS

DWORD wAddr[in] : 센서의 Address 값

DWORD* wData[in, out] : 해당 Address Value 값

RETURN VALUE

성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE

HW_IIC_Read_SlaveAddr

HW_IIC_Read 와 같은 기능을 하는 함수. Slave Address 를 인자로 넣어 사용할 수 있다.

Remarks

HW_IIC_Read 함수는 Default SlaveAddr을 ini로부터 Read하여 이 Slave Addr에 해당하는 정보를 Read한다. 본 함수는 ini File에서 읽어오는 것이 아닌 개발자가 함수 인자로 Slave Addr을 입력하여 센서 기억소자(Register) 정보를 읽을 수 있는 함수이다.

BOOL HW_IIC_Read_SlaveAddr(DWORD wAddr, DWORD *wData, WORD wSlaveAddr)

PARAMETERS

DWORD wAddr[in] : 센서의 Address 값

DWORD* wData[in, out] : 해당 Address Value 값

WORD wSlaveAddr[in] : Slave Address(사용자가 선택 가능함)

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	36 of 73

RETURN VALUE

성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE**HW_IIC_Read_SelMode**

HW_IIC_Read 와 같은 기능을 하는 함수. Mode를 인자로 넣어 사용할 수 있다.

Remarks

HW_IIC_Read 함수는 Default Mode를 ini로부터 Read하여 이 Slave Addr에 해당하는 정보를 Read한다. 본 함수는 ini File에서 읽어오는 것이 아닌 개발자가 함수 인자로 모드를 입력하여 센서 기억소자(Register)정보를 읽을 수 있는 함수이다.

BOOL HW_IIC_Read_SelMode(INT mode, DWORD wAddr, DWORD *wData);

PARAMETERS

INT mode[in]: I2C Mode(0:I2C_A8_D8, 1:I2C_A8_D16, 2: I2C_A16_D8, 3:I2C_A16_D16)

DWORD wAddr[in] : 센서의 Address 값

DWORD* wData[in, out] : 해당 Address Value 값

RETURN VALUE

성공시 TRUE, 실패시 FALSE 리턴

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	37 of 73

HW_IIC_Read_BULK

인자로 들어가는 Address 및 Data를 버퍼에 저장하는 함수

Remarks

사용 Sensor가 Bulk Mode를 지원하는지 여부를 확인하여야 한다.

BOOL HW_IIC_Read_BULK(INT read_cnt, CHAR* addr_buf, CHAR* read_buf);

PARAMETERS

INT read_cnt[in]: Input Read Count(최대 100Byte)

CHAR* addr_buf[in, out] : Count만큼 Read한 Address Buffer

CHAR* read_buf[in, out] : Count만큼 Read한 Address에 대한 Data Buffer

RETURN VALUE

성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE**HW_IIC_Write_BULK**

CHAR형의 버퍼를 센서에 Write하는 기능을 한다. (사용 센서가 해당 기능을 지원하는지 여부 확인이 필요함)

Remarks

사용 Sensor가 Bulk Mode를 지원하는지 여부를 확인하여야 한다.

BOOL HW_IIC_Write_BULK(CHAR* tmp_buf);

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	38 of 73

PARAMETERS

CHAR* write_buf[in] : write할 buffer

RETURN VALUE

성공시 TRUE, 실패시 FALSE 리턴

EXAMPLE**HW_StartDevice**

장치 시작 함수

Remarks

이 함수는 다음과 같은 flow를 가진다

1. interrupt 및 control register을 초기화한다.
2. Video 입력의 해상도를 set
3. Frame Data 수집을 시작

BOOL HW_StartDevice();**PARAMETERS**

None

RETURN VALUE

FLOW 전부 성공시 TRUE RETURN, 하나라도 실패시 FALSE 리턴

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	39 of 73

GetID_USB4

V4 계열 보드 USB ID를 얻어오는 함수

int GetID_USB4();

USB ID를 얻어오는 함수

PARAMETERS

None

RETURN VALUE

V4의 Board ID를 Return한다.

[HVS_HW]

HW_INDEX 의 Value(8000, 8001, 8002, 8003 이 아닐 경우 0이 리턴됨)

EXAMPLE**GetIndex_USB4**

V4 계열 보드 USB Index를 얻어오는 함수

int GetIndex_USB4(LPSTR UsbBoardID_V4);

PARAMETERS

"8000", "8001", "8002", "8003" 등의 String을 넘겨준다.

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	40 of 73

RETURN VALUE

V4의 Board Index를 Return한다.

(0, 1, 2, 3)만약 ID에 해당하는 Index가 없을 때 0을 Default로 사용

EXAMPLE**OnHWDlg**

```
void OnHWDlg(CWnd *pParentWnd);
```

PARAMETERS

pParentWnd[in]: 부모 윈도우의 핸들

RETURN VALUE

없음

EXAMPLE**HW_RegisterSet_R3**

```
BOOL HW_RegisterSet_R3(LPSTR lpcommand, UINT nHWMMode = 0, UINT nGpioLowHigh = 0)
```

PARAMETERS

lpcommand[in]: 사용자 정의 Key

EX) [AE_SETTING] -> Command의 인자로 들어간다.

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	41 of 73

0x3601 0x11

[END]

nHWMODE[in]: OS&P를 사용하는 경우 1, 사용하지 않는 경우 0을 인자로 입력해 준다.

nGpioLowHigh[in]: 현재 사용하지 않는 인자

RETURN VALUE

None

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	42 of 73

PlatformMemory APIs

OverView

void	OnPlatformMemoryCreate(void* pMemory)
Platform_info*	Platform_Info* GetPlatformInfo();
Module_Info*	GetModuleInfo();
void	GetPlatformVersion(TCHAR* pBuffer, TCHAR* pFileName);

OnPlatformMemoryCreate

주의사항: Test Item 개발자는 위 함수를 절대로 사용하지 말 것(Main Platform 관련 함수)

Platform 관련 Memory 생성 함수.

Memory class는 test item의 Interface 역할을 하는 멤버 변수 및 멤버 함수가 정의.

void OnPlatformMemoryCreate(void* pMemory);

PARAMETERS

pMemory: void형 Pointer(Memory 접근 인자)

RETURN VALUE

None

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	43 of 73

GetPlatformInfo

Platform 관련 정보를 가져오는 함수. (자세한 사항은 #Appendix A : Structure 참고)

Platform_Info* GetPlatformInfo();

PARAMETERS

None

RETURN VALUE

Platform_Info* (Appendix A : Structure 참고)

EXAMPLE

GetModuleInfo

Test, Module 관련 정보를 가져오는 함수. (자세한 사항은 #Appendix A: Structure 참고)

현재 Barcode, dll Test 누적 시간에 대한 정보를 얻어온다. Barcode는 핸들러로부터 TCP/IP Socket 통신을 이용해 받아오는 정보

Module_info* GetModuleInfo();

PARAMETERS

None

RETURN VALUE

Module_Info* (#Appendix A : Structure 참고)

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	44 of 73

EXAMPLE**GetPlatformVersion**

Platform 버전 정보를 가져온다.

```
void GetPlatformVersion(TCHAR* pBuffer, TCHAR* pFileName);
```

PARAMETERS

pBuffer: 버전 정보를 담을 Buffer

pFileName: Root 경로의 File Name

RETURN VALUE

None

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	45 of 73

UserProtocol APIs

OverView

void	OnProtocolInit(void *pData);
MES_Info*	GetMesInfo();
Barcode_info*	GetBarCodeInfo();
CowellData*	GetCowellData();
MesTimeInfo*	GetMesTimeInfo();
void	OnLightSource(int nChannel);
BOOL	MoveCylinder(BOOL bMove);
void	MoveCylinderComplete();
BOOL	GetMoveCylinderComplete();
void	StopFocusTestComplete();
BOOL	GetIsStopFocusTest();
BOOL	WaitFocusConeLoad(UINT nTimeOver_ms);
BOOL	MoveFocusMotor(float fDegree, UINT nTimeOver_ms);
BOOL	MoveFocusZ(BOOL bisUp, UINT nTimeOver_ms);
BOOL	MoveFocusChart(int nXpixel, int nYpixel, UINT nTimeOver_ms);
BOOL	RequestDataToHandler(int nMode, UINT nTimeOver_ms, double *dRet);

OnProtocolInit

Protocol 관련 정보 초기화 함수

```
void OnProtocolInit(void *pData);
```

PARAMETERS

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	46 of 73

pData[in, out]: void형 Pointer

RETURN VALUE

None

EXAMPLE**GetMesInfo**

MES 정보(Module 물류 관리 정보)를 얻어오는 함수(자세한 사항은 #Appendix A : Structure 참고)

MES_Info* GetMesInfo();

PARAMETERS

None

RETURN VALUE

MES_Info* : MES 정보 관련 구조체([#Appendix A : Structure](#) 참고)

EXAMPLE**GetBarcodeInfo**

Barcode 관련 정보를 얻어오는 함수(자세한 사항은 [#Appendix A : Structure](#) 참고)

Barcode_info* GetBarcodeInfo();

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	47 of 73

PARAMETERS

None

RETURN VALUE

Barcode_info* : Barcode 정보 관련 구조체(#Appendix A : Structure 참고)

EXAMPLE**GetMesTimeInfo**

Mes 시간 정보 얻어오는 함수(현재 time 정보만 얻어옴)

MesTimeInfo* GetMesTimeInfo();**PARAMETERS**

None

RETURN VALUE

MesTimeInfo*: MES 시간 정보

EXAMPLE**OnLightSource**

광원 Control을 위해 지정된 Packet Send 함수

void OnLightSource(int nChannel);

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	48 of 73

PARAMETERS

nChannel: Channel 정보

RETURN VALUE**EXAMPLE****MoveCylinder**

Cylinder Control을 위한 Packet Send 함수

BOOL MoveCylinder(BOOL bMove);

PARAMETERS

bMove[in]: TRUE or FALSE

RETURN VALUE**EXAMPLE****MoveCylinderComplete**

Cylinder 이동 완료 여부 지정하는 함수임(내부적으로 Flag가 FALSE -> TRUE로 바뀜)

void MoveCylinderComplete();

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	49 of 73

PARAMETERS

None

RETURN VALUE

None

GetMoveCylinderComplete

Cylinder 이동이 완료되었는지 여부 확인하는 함수

Remarks

Cylinder 이동이 완료되기 전에 다른 동작을 하는 경우 원하지 않는 결과를 초래할 가능성 있음. Cylinder 이동중에 Capture를 하는 경우를 예로 들 수 있다.

BOOL GetMoveCylinderComplete();**PARAMETERS**

None

RETURN VALUE

Cylinder 이동 완료시 TRUE, 정지 혹은 이동중인 경우 FALSE return

EXAMPLE**StopFocusTestComplete**

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	50 of 73

Focus test 루틴 정지 완료 함수(AutoFocus에서 TEST STOP시 항상 이 함수를 호출하여야 한다.)

void StopFocusTestComplete();

PARAMETERS

None

RETURN VALUE

None

GetIsStopFocusTest

Handler로부터 Stop Command가 왔는지 여부 확인하는 함수

BOOL GetIsStopFocusTest();

PARAMETERS

None

RETURN VALUE

STOP시 TRUE, 그렇지 않을 경우 FALSE RETURN

EXAMPLE

WaitFocusConeLoad

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	51 of 73

Cone Load 완료되기까지 대기하는 함수

BOOL WaitFocusConeLoad(UINT nTimeOver_ms);

PARAMETERS

nTimeOver_ms[in]: TimeOut 인자. [ms] 단위

RETURN VALUE

Cone Load 정상적으로 완료시에는 TRUE, TimeOut날 경우 FALSE RETURN

EXAMPLE

```
bResult = WaitFocusConeLoad( m_nMotionTimeOut );
if(bResult == FALSE)
{
    nResultCode = R_ERROR_CANNOTDO;
    ListLog("Focus cone error");
    return nResultCode;
}
else
{
    ListLog("Cone load complete");
}
```

MoveFocusMotor

Focus Cone 회전 제어 함수

BOOL MoveFocusMotor(float fDegree, UINT nTimeOver_ms);

PARAMETERS

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	52 of 73

fDegree[in]: 회전 각도

nTimeOver_ms[in]: TimeOut 인자. [ms] 단위

RETURN VALUE

성공시 TRUE, 실패시 FALSE

EXAMPLE

```
if(fAngle == 0.0f) return TRUE;

if(bBackrush == TRUE)
{
    backrush_offset = m_fBackrushAngle;
    if(fAngle > 0.0f) angle = fAngle + m_fBackrushAngle; //양수로 backrush 만큼 더 진행
    else                angle = fAngle - m_fBackrushAngle; //음수로 backrush 만큼 더 진행

    //본 각도
    bResult = MoveFocusMotor(angle, fTimeout);
    if(!bResult) goto _end;

    //backrush 각도
    if(fAngle > 0.0f) bResult = MoveFocusMotor( m_fBackrushAngle* -1.0f, fTimeout);
    else                bResult = MoveFocusMotor( m_fBackrushAngle, fTimeout);
    if(!bResult) goto _end;
}
}
```

MoveFocusZ

Cone Lift Up / Down 제어 함수

BOOL MoveFocusZ(BOOL bisUp, UINT nTimeOver_ms);

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	53 of 73

PARAMETERS

bisUp[in]: TRUE일 경우 Cone Up / FALSE일 경우 Cone Down

nTimeOver_ms[in]: TimeOut 인자. [ms] 단위

RETURN VALUE

성공시 TRUE, 실패시 FALSE

EXAMPLE**MoveFocusChart**

Moving Chart 위치 제어 함수

MoveFocusChart(int nXpixel, int nYpixel, UINT nTimeOver_ms);

PARAMETERS

nXpixel[in]: X Axis Pixel Value

nYpixel[in]: Y Axis Pixel Value

nTimeOver_ms[in]: TimeOut 인자. [ms] 단위

RETURN VALUE

성공시 TRUE, 실패시 FALSE

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	54 of 73

RequestDataToHandler

Handler에게 Laser, Vision 등의 정보를 요청하는 함수

RequestDataToHandler(int nMode, UINT nTimeOver_ms, double *dRet);

PARAMETERS

nMode[in]: Laser: 0 Vision X: 1 Vision Y: 2 Vision Angle: 3

nTimeOver_ms[in]: TimeOut 인자. [ms] 단위

*dRet[in, out]: Result Data

RETURN VALUE

성공시 TRUE, 실패시 FALSE

EXAMPLE

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	55 of 73

#Appendix A : Structure

Config

Data Type	Variable Name	Explanation
BoardControl(structure)	Config_Board	Board Control Structure(inner)
MCUBoardControl(Structure)	Config_MCU_Board	MCUBoardControl(OS&P) Structure(inner)
void*	pHVSDDevice	Reserved Pointer(Related HVS Device)
void*	pUserDevice	Reserved Pointer(Related User Device)
HWND	m_hList	ListOut Dialog Window Handle
int	m_nListItem	ListOut Dialog Control ID Num
int	m_nIndex	List Out Line Count Index
BOOL	bTestItemTerminate	Test Item Terminate Flag Variable
BOOL	bIsTestSequenceStop	It is not use until now.
HWND	hTCP_hWnd	Window Handle(For TCP Event)
UINT	nTCP_Message	TCP USER Window Message
YUV_Table	YUVTable	YUV_Table structure
void*	pTestBuffer	Reserved User Create Memory reference Pointer
HWND	hDraw	It is not use until now.
DrawImage	Config_Image	DrawImage Structure
_DllMemory	DllMemory	_DllMemory Structure
CvImage	cvImage	CvImage Structure

TestInfo

Data Type	Variable Name	Explanation
-----------	---------------	-------------

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	56 of 73

HVS_HW (Structure)	hwSet	Reference HVS_HW Structure
HVS_MCU (Structure)	MCUSet	Reference HVS_MCU Structure
UINT	nImage_Width	Image Width
UINT	nImage_Height	Image Height
UINT	nImage_Width_Prev	Preview Width
UINT	nImage_Height_Prev	Preview Height
IpImage*	*imgIntP	Normal Size Image Buffer Pointer
IpImage*	*imgIntP_Prev	Not Normal Size Image Buffer Pointer
void*	pRev1	Reserved allocated memory address1
void*	pRev2	Reserved allocated memory address2
void*	pRev3;	Reserved allocated memory address3

HVS_HW

Data Type	Variable Name	Explanation
BOOL	bInitOK	HW Initial(V4, V5) OK or Not
UINT	nBoardType	BoardType(1: V4I 2:V4P 3: V5U 4: V5U_R2)
UINT	nUSB_BoardID	Usb Board ID
UINT	nPCI_BoardID	PCI Board ID
BOOL	bPowerOn[POWER_CH]	Power On Pin Select(total 6 Pin)
BOOL	bPowerOff[POWER_CH]	Power Off Pin Select(total 6 Pin)
UINT	nPowerOrder[POWER_CH]	Power Order of priority
UINT	nPowerDelay[POWER_CH]	Power deley between pin to pin
float	fPowerVolt[POWER_CH]	Voltage Value(each Pin)
float	fPowerIO_Volt	IO Voltage Value

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	57 of 73

float	fMClock	Master Clock Speed Value
BOOL	bSignalReset	TRUE: Reset High, FALSE: Reset Low
BOOL	bSignalEnb	TRUE: Enable High, FALSE: Reset Low
UINT	nMipiType	mipiType (0: MIPI 8Bit , 1: MIPI 16Bit 3: MIPI 32Bit , 9: parallel Interface Type)
UINT	nMipiMode	Mipi Lane select (0: 1Lane 1: 2Lane 3: 3Lane 7: 4Lane)
UINT	nColorOrder_Mode	Color Order 0:YCbYCr, 1:YCrYCb, 2:CbYCrY, 3:CrYCbY 0:BGGR, 1:RGGG, 2:GBRG, 3:GRBG, 4:BlackWhite
WORD	wSlaveAddr	Slave Address
UINT	nI2C_Mode	0: 8Bit_8Bit 1: 8Bit_16Bit 2: 16Bit_8Bit 3: 16Bit_16Bit
UINT	nDataFormat	0: YUV 1: 8Bit Bayer 2: 10Bit Bayer 3: 12Bit Bayer

HVS_MCU

Data Type	Variable Name	Explanation
BOOL	bPowerOn[MCU_POWER_CH]	HVS OS&P Power On Pin Select(total 6 Pin)
float	m_fMCUPowerVolt[MCU_POWER_CH]	B HVS OS&P Voltage Value(each Pin)
float	fMClock	Master Clock Speed Value
BOOL	bMCUSignalReset	TRUE: Reset High, FALSE: Reset Low
BOOL	bMCUSignalEnb	TRUE: Enable High, FALSE: Reset Low
WORD	dwMCUSlaveAddr	Slave Address

Platform_info

Data Type	Variable Name	Explanation
BOOL	bStopButtonFlag	Platform State (TRUE: STOP, FALSE: WORKING)

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	58 of 73

BOOL	bOverrideMode	Test Mode: Override Mode or Not
BOOL	bSimulationMode	Test Mode : Simulation Mode or Not
BOOL	bSimulImageSave	Test Mode : Simulation Image Save or Not
BOOL	bSimulNoneImage	Test Mode : Simulation None Image or Not
BOOL	bSimulPath	Test Mode : 0 : folder , 1 : default image
TCHAR	szFolderImagePath[MAX_PATH]	Simulation Image Folder Path
TCHAR	szDefaultImagePath[MAX_PATH]	Simulation Image Path
UINT	nSiteIndex	0: A Para, 1:B Para, 2:C Para, 3: D Para
UINT	nLoginLevel	Config : LEV_OPERATOR 0, LEV_MAINTENANCE 1, LEV_ENGINEER 2
TCHAR	szRoot[MAX_PATH]	Platform exe : Platform.exe root path
TCHAR	szModelName[MAX_PATH]	Config : Model Name
TCHAR	szMachineName[MAX_PATH]	Config : Machine Name
TCHAR	szLineName	Config : Line Name
TCHAR	szSimulationPath	Config : Simulation Path;
TCHAR	szPlatformLogPath	Config : Platform Log Path
TCHAR	szProductionPath	Config : Production Log Path
TCHAR	szEnginnerPath	Config : Enginner Log Path

Module_info

Data Type	Variable Name	Explanation
TCHAR	SzN_Num[18]	Platform State (TRUE: STOP, FALSE: WORKING)
float	fModuleTime	Test Time(from test start ~ test end. Not including cycle end)

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	59 of 73

MES_Info

Data Type	Variable Name	Explanation
UINT	nTestMode	0: Production Mode 1: Engineer Mode 2: Retest Mode
UINT	nOnOffLine	OnLine or OffLine
UINT	nBCDChecksum	Use Checksum or don't use Checksum
BOOL	bFocusMode_FCheck	F Check Use
BOOL	bFocusMode_ThroughFocus	ThroughFocus Use
TCHAR	szLot_ID[MAX_PATH]	Lot ID Name
TCHAR	szEquipment[MAX_PATH]	Equipment Name
TCHAR	szFactory[MAX_PATH]	Factory Name
TCHAR	szArea[MAX_PATH]	Area Name
TCHAR	szLine[MAX_PATH]	Line Name
TCHAR	szStation[MAX_PATH]	Station Name
TCHAR	szOperation[MAX_PATH]	Operation Name
TCHAR	szSubEquipment[MAX_PATH]	SubEquipment Name
TCHAR	szProductspec[MAX_PATH]	Productspec Name
TCHAR	szEventuser[MAX_PATH]	EventUser Name
TCHAR	szImagefilepath[MAX_PATH]	ImageFilePath
TCHAR	szEdcra data[MAX_PATH]	Edcra Data
TCHAR	szReasoncodetype[MAX_PATH]	ReasoncodeType
TCHAR	szReasoncode[MAX_PATH]	ReasonCode
TCHAR	szSWversion[MAX_PATH]	SW Version
TCHAR	szServerAddr[MAX_PATH]	ServerAddr

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	60 of 73

barcode_info

Data Type	Variable Name	Explanation
int	nBCD_Length	Barcode length
int	nBCD_PosNo_0	Barcode Info
int	nBCD_PosNo_2	Barcode Info
int	nBCD_PosNo_1	Barcode Info
int	nBCD_PosNo_3	Barcode Info
int	nBCD_PosNo_4	Barcode Info
BOOL	bBCD_Checksum	Barcode CheckSum
BOOL	bBCD_Check_0	Barcode Info
BOOL	bBCD_Check_1	Barcode Info
BOOL	bBCD_Check_2	Barcode Info
BOOL	bBCD_Check_3	Barcode Info
BOOL	bBCD_Check_4	Barcode Info
TCHAR	szBarCode[20]	Barcode Info
TCHAR	szBCD_Char_0[MAX_PATH]	Barcode Type1(User Define)
TCHAR	szBCD_Char_1[MAX_PATH]	Barcode Type2(User Define)
TCHAR	szBCD_Char_2[MAX_PATH]	Barcode Type3(User Define)
TCHAR	szBCD_Char_3[MAX_PATH]	Barcode Type4(User Define)
TCHAR	szBCD_Char_4[MAX_PATH]	Barcode Type5(User Define)

#Appendix B : HVS Data Define Constant

//Hyvision data structure

#define RESULT_PASS 0

#define RESULT_FAIL 1

#define HW_MODE_NORMAL 0

#define HW_MODE_USE_OS_MCU 1

#define HW_MODE_USE_MDMI 2

#define GPIO_LOW 0

#define GPIO_HIGH 1

#define POWER_CH 6

#define HVS_V4I 1

#define HVS_V4P 2

#define HVS_V5U 3

#define HVS_V5R 4

#define POWER_CH 6

#define MCU_POWER_CH 4

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	62 of 73

```

#define GPIO_CH          4
#define GPIO_1           0
#define GPIO_2           1
#define GPIO_3           2
#define GPIO_4           3

#define SIGNAL_CH        2
#define SIGNAL_RESET     0
#define SIGNAL_ENB       1

//I2C_Mode
#define I2C_A8_D8        0
#define I2C_A8_D16       1
#define I2C_A16_D8       2
#define I2C_A16_D16      3

//YUV color order
#define YCbYCr           0
#define YCrYCb           1
#define CbYCrY           2
#define CrYCbY           3

//Raw color order
#define BGGR              0
#define RGGGB            1

```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	63 of 73

HVS New Platform USER GUIDE

Product Name: **HVS API Programming(2.0)**

```
#define GBRG          2
#define GRBG          3
#define BlackWhite    4
```

//Input Image data format

```
#define FORMAT_YUV      0
#define FORMAT_RAW8     1
#define FORMAT_RAW10    2
#define FORMAT_RAW12    3
```

//input mipi data lane

```
#define MIPI_1LANE      0
#define MIPI_2LANE      1
#define MIPI_4LANE      2
```

```
#define INPUT_DATA_MIPI  0
#define INPUT_DATA_PARALLEL 1
```

//IIC mode

```
#define BIT8_BIT8        0
#define BIT8_BIT16       1
#define BIT16_BIT8       2
#define BIT16_BIT16      3
```

//DataMaxSize

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	64 of 73

HVS New Platform USER GUIDE

Product Name: **HVS API Programming(2.0)**

#define DATA_MAX_X 4208

#define DATA_MAX_Y 3200

#define MAXSTR 500

#define MAX_DEVICE 4

#define V4BOARD 0

#define V4PBOARD 1

#define V4IBOARD 2

#define HVS_V4I 1

#define HVS_V4P 2

#define HVS_V5U 3

#define HVS_V5R 4

#define POWER_CH 6

#define MCU_POWER_CH 4

#define GPIO_CH 4

#define GPIO_1 0

#define GPIO_2 1

#define GPIO_3 2

#define GPIO_4 3

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	65 of 73

HVS New Platform USER GUIDE

Product Name: **HVS API Programming(2.0)**

```
#define SIGNAL_CH          2
#define SIGNAL_RESET      0
#define SIGNAL_ENB        1
```

//I2C_Mode

```
#define I2C_A8_D16        1
#define I2C_A16_D8       2
#define I2C_A8_D8        0
#define I2C_A16_D16      3
```

//YUV color order

```
#define YCbYCr           0
#define YCrYCb           1
#define CbYCrY           2
#define CrYCbY           3
```

//Raw color order

```
#define BGGR             0
#define RGGB             1
#define GBRG             2
#define GRBG             3
#define BlackWhite       4
```

//Input Image data format

```
#define FORMAT_YUV        0
#define FORMAT_RAW8       1
```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	66 of 73

HVS New Platform USER GUIDE

Product Name: **HVS API Programming(2.0)**

#define FORMAT_RAW10 2

#define FORMAT_RAW12 3

//input mipi data lane

#define MIPI_1LANE 0

#define MIPI_2LANE 1

#define MIPI_4LANE 2

#define INPUT_DATA_MIPI 0

#define INPUT_DATA_PARALLEL 1

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	67 of 73

#Appendix C : Sample Project

```
//Test item function
extern "C" __declspec(dllexport)
int OnTest(UINT nTestNumber, void *pData)
{
    int nResult = RESULT_FAIL;
    gTest.InitTest(pData); //초기화 함수(수정하지 말것)
    nResult = gTest.StartTest(nTestNumber); //Test 루틴 구현
    return nResult;
}
```

```
//Show spec dialog
extern "C" __declspec(dllexport)
void* OnSpecDlg(CWnd *pParentWnd, void *pData)
{
    gTest.InitTest(pData); //초기화 함수(수정하지 말것)
    gTest.m_SpecDlg.DoModal();
    return NULL;
}
```

```
int CTest::InitTest(void *pData)
{
    int nResultCode = RESULT_PASS;
    //경로 정보등을 받아 온다.

    //HVS:수정하지 말것.
    SetHVS_Data(pData);
    GetTestInfo(&m_TestInfo);
    SetRepeatTest_IsStop(FALSE);
    //~HVS

    return nResultCode;
}
```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	68 of 73

```
}
int CTest::StartTest(UINT nTestNo)
{
    int nResultCode = RESULT_FAIL;

    Before_TestProcedure(nTestNo);
    nResultCode = V5U_TestProcedure();

    if(nResultCode==19)
        SetTestSequenceStop(TRUE);

    After_TestProcedure();

    return nResultCode;
}
void CTest::Before_TestProcedure(UINT _nTestNo)
{
    m_TestNo = _nTestNo;

    GetTestInfo(&m_TestInfo);

    CreateFolder(m_pPlatformInfo->szEnginnerPath, m_pPlatformInfo->szProductionPath);

    m_Image_Width = m_TestInfo.nImage_Width;
    m_Image_Height= m_TestInfo.nImage_Height;

    //CreateBuffer
    m_pBuffer = GetImageBuffer(m_Image_Width, m_Image_Height);
    Create_Test_Common_Variable(m_Image_Width, m_Image_Height, m_pBuffer);
    m_pCommonTestBuf = (pTest_Common_Variable)m_pBuffer->pUserBuffer0;

    ReadSpec();
}
int CTest::V5U_TestProcedure()
{
    CString strTemp;
    UINT nDummyCount = m_nDummyCount;
    UINT nCaptureCount = m_nCaptureCount;
    UINT nDataFormat    = m_TestInfo.hwSet.nDataFormat;
    UINT nColorOrder   = m_TestInfo.hwSet.nColorOrder_Mode;
```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	69 of 73

```
WORD *m_pWordTemp;
BYTE *m_pBYTETemp;
m_pWordTemp    = new WORD[m_Image_Width * m_Image_Height * 4];
m_pBYTETemp    = new BYTE[m_Image_Width * m_Image_Height * 4];

memset(m_pWordTemp, 0, sizeof(WORD)*m_Image_Width * m_Image_Height * 4);
memset(m_pBYTETemp, 0, sizeof(BYTE)*m_Image_Width * m_Image_Height * 4);

if(m_bSimulationMode)
{
    //저장 후 구현할 것
    ListOut("Simulation Mode Start");
    if(m_bUseBMP==FALSE)
    {
        ListOut("Load Raw Image!");
        LoadRawImageToBuffer(m_pCommonTestBuf->m_pRawBuffer, m_Image_Width, m_Image_Height, m_TestInfo.hwSet.nDataFormat);
        //m_pDoubleRawBuffer

        Convert_Bit10_To_Bit8(m_pCommonTestBuf->m_pRawBuffer, m_pBYTETemp, m_Image_Width, m_Image_Height);
        Convert_RAW_To_RGB_BYTE(m_pBYTETemp, m_pCommonTestBuf->m_pBmpBuffer, m_Image_Width, m_Image_Height,
            m_TestInfo.hwSet.nColorOrder_Mode);
    }
    else
    {
        //Load Image
        ListOut("Load Bmp Image!");
        OPENFILENAME        OFN;
        TCHAR                lpsfrFile[MAX_PATH]={0,};
        memset(&OFN, 0, sizeof(OPENFILENAME));
        OFN.lStructSize = sizeof(OPENFILENAME);
        OFN.hwndOwner = NULL;
        OFN.lpstrFilter = "BMP FILE(*.*)\\0*.bmp\\0";
        OFN.lpstrFile = lpsfrFile;
        OFN.nMaxFile = MAX_PATH;
        OFN.nMaxFileTitle = MAX_PATH;

        if(!GetOpenFileName(&OFN))
        {
            ListOut("Error Get file name!!!");
        }
    }
}
```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	70 of 73

```
        }
        else
            LoadImage(m_pCommonTestBuf->m_pBmpBuffer, OFN.lpszFile);
    }
}
else
{
    if(!GetImage_General(m_pCommonTestBuf->m_pCaptureBuffer, m_Image_Width, m_Image_Height, m_TestInfo.hwSet.nDataFormat,
        nDummyCount,nCaptureCount))
    {
        ListOut("Image Capture Fail1");
        return R_FAIL_CAPTURE;
    }
    Convert_V5DataBit10_To_Bit10(m_pCommonTestBuf->m_pCaptureBuffer, m_pWordTemp, m_Image_Width, m_Image_Height);
    Convert_WORD_To_BYTE_10Bit(m_pWordTemp, m_pCommonTestBuf->m_pRawBuffer, m_Image_Width, m_Image_Height);

    Convert_Image(nDataFormat, nColorOrder, m_pCommonTestBuf->m_pCaptureBuffer, m_pCommonTestBuf->m_pBmpBuffer, m_Image_Width,
        m_Image_Height);

    //////////////////////////////////////
    //save Capture image
    if(m_bSaveCaptureImage==TRUE)
    {
        SaveCapturedRawImage(m_pCommonTestBuf->m_pRawBuffer,m_Image_Width, m_Image_Height,m_pCommonTestBuf-
            >strTestStartTime,nDataFormat);
    }
    if(m_bSaveBmpImage==TRUE)
    {
        SaveCapturedBmpImage(m_pCommonTestBuf->m_pBmpBuffer,m_Image_Width, m_Image_Height,m_pCommonTestBuf-
            >strTestStartTime,nDataFormat);
    }
    ListOut("Image Capture Complete");
}
Convert_BYTE_To_double(m_pCommonTestBuf->m_pRawBuffer, m_pCommonTestBuf->m_pDoubleRawBuffer, m_Image_Width, m_Image_Height,
    HIGH_LOW_SUBDATA_VERMONT);
//ColorCal
Convert_RGB_To_YUV_To_YRGB_BYTE( m_pCommonTestBuf->m_pBmpBuffer, m_pCommonTestBuf->m_pYBmpBuffer, m_Image_Width, m_Image_Height);

delete [] m_pWordTemp;
delete [] m_pBYTETemp;
return RESULT_PASS;
```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	71 of 73

```
}

int CTest::After_TestProcedure()
{
    DrawImage(m_pCommonTestBuf->m_pBmpBuffer);

    return RESULT_PASS;
}
BOOL CTest::DrawImage(LPBYTE pImgBuf)
{
    if(pImgBuf == NULL)
    {
        ListOut("Draw Image Input Para is NULL!!!");
        return FALSE;
    }
    //Byte image 입력
    Input_Image_Buffer(pImgBuf, m_Image_Width, m_Image_Height, FALSE);

    CvFont Font;
    cvInitFont(&Font, CV_FONT_HERSHEY_SIMPLEX, 1.0f, 1.0f);
    char szText[50];
    sprintf(szText, "%s", "Capture Image Draw");
    cvPutText(m_TestInfo.imgIntP, szText, cvPoint(50, 100), &Font, CV_RGB(255, 255, 0));

    SYSTEMTIME st;
    GetLocalTime(&st);

    Input_cvImage(m_TestInfo.imgIntP);
    Output_Image();

    return TRUE;
}
int CTest::ReadSpec( )
{
    CString strPath = GetTestItemSpecPath();
    int nResultCode = RESULT_PASS ;

    HWND hDlg = m_SpecDlg.GetDlgHandle();
    m_bSimulationMode = GetDlgItemInt(hDlg, IDC_CHECK_SIMULATION_ON, NULL, FALSE);
}
```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	72 of 73


```

        m_bSimulationMode      = INIRead_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "SIMULATION_MODE_ON", FALSE); //20130713
        m_bUseBMP              = INIRead_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "USE_BMP", FALSE);;
        m_bSaveCaptureImage    = INIRead_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_SAVE_ON_CAPTURE", FALSE); //20130715
        m_bSaveBmpImage        = INIRead_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_SAVE_ON_BMP", FALSE); //20130715
        m_bSaveResultImage     = INIRead_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_SAVE_ON_RESULT", FALSE); //20130715

        m_nDummyCount          = INIRead_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_DUMMY_COUNT", 1);
        m_nCaptureCount        = INIRead_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_CAPTURE_COUNT", 1);
        return nResultCode;
    }
    int CTest::SaveSpec()
    {
        CString strPath = GetTestItemSpecPath();

        int nResultCode = RESULT_PASS;

        INIWrite_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "SIMULATION_MODE_ON", m_bSimulationMode);
        INIWrite_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "USE_BMP", m_bUseBMP);
        INIWrite_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_SAVE_ON_CAPTURE", m_bSaveCaptureImage);
        INIWrite_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_SAVE_ON_BMP", m_bSaveBmpImage);
        INIWrite_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_SAVE_ON_RESULT", m_bSaveResultImage);

        INIWrite_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_DUMMY_COUNT", m_nDummyCount);
        INIWrite_File((LPSTR)(LPCTSTR)strPath, "OPERATION", "IMAGE_CAPTURE_COUNT", m_nCaptureCount);
        return nResultCode;
    }
}

```

Document Number	Template Title:	Written By / Owner	Revision Code	Page
20140807_PA	HVS Platform API	HVS	V1.0	73 of 73