



Imagination PXR800 & Visual Basic[®]

Introduction

To assist the Visual Basic programmer using an Imagination PXR800, CyberOptics Semiconductor has provided a set of tools to access the PXR800 libraries. These tools allow a Visual Basic application to use the PXR800. Samples are also provided.

This document is a supplement to the *PXR800 Frame Grabber User's Guide* and provides detailed information for developing Visual Basic applications that use the PXR800. It is assumed that the programmer has reviewed the *PXR800 Frame Grabber User's Guide* and is already familiar with Visual Basic programming. The *PXR800 Frame Grabber User's Guide* can be downloaded from www.imagination.com.

Programming with Visual Basic

The PXR800 libraries (DLLs) were designed to make the function calls as uniform as possible. Before you start programming in Visual Basic, you could look at the Visual Basic function definitions in the files PXRFRAME.BAS and PXR800API.BAS.

Note: To use the PXR800 in Visual Basic applications, you must have installed version 1.4 or later of the PXR800 libraries. These libraries are included with the PXR800_VB.EXE package.

There are a few things you should keep in mind when using Visual Basic with the DLL functions:

Accessing frame data

Visual Basic doesn't use pointers, so you must use the functions `GetPixel()`, `GetColumn()`, `GetRectangle()`, and `GetRow()` to access the data in a frame. Visual Basic includes a `Byte` type, which is equivalent to the `unsigned char` type that the DLLs expect for buffers. To pass a buffer to the DLL, just pass the first element of your declared `Byte` array. That is, by providing element (0), or element (0, 0). For example, to get a rectangle of 8 bpp pixels from a frame, one can use code such as:

```
Buff1(7500) As Byte
I = FRAMEVB_GetRectangle(Frame, Buff1(0), 10, 10,
100, 75);
```

Including .BAS Files

You must include the appropriate .BAS file in all projects you build using the PXR800 library functions. The .BAS files include all the declarations you'll need to work with the DLLs. To display your images, you can include PXR800_Display.BAS.

Caution Do not use the **End** button in the Visual Basic development environment to terminate your application. The End button terminates a program immediately, without executing the Form_Unload function or any other functions. If you use the **End** button to exit a program, you might need to exit and restart Visual Basic to free any frame grabbers that your program allocated.

Displaying Video in Visual Basic Applications

The PXR800 software for Visual Basic includes a Video Display Library – PXR800_DISPLAY.DLL – that helps to display captured images in a window. This library can be accessed through the PXR800_DISPLAY.BAS file. The Video Display Library is not required to use the PXR800 API libraries or the PXR800 Frame libraries – it merely provides some helpful functions.

Using the Video Display Library

The helper Video Display Library is a simple tool for displaying video images in a window. Since it is a standard DLL, it can be used with Visual Basic, C, and other languages that can call DLLs. The Video Display Library supports only one operation: copying an arbitrary rectangle of an image frame onto an arbitrary rectangle of a window's client area. There are two functions that are needed for this purpose:

```
IMG_VB_PaintDisplayWindow (ByVal hdc As Long, ByVal  
    PXR_Frame As Long, ByVal x As Long, ByVal y As  
    Long, ByVal dx As Long, ByVal dy As Long)
```

```
IMG_VB_SizeDisplayWindow (ByVal x As Long, ByVal y  
    As Long, ByVal dx As Long, ByVal dy As Long)
```

To incorporate the Video Display Library into your programs, you will need these files:

Video Display Files

PXR800_Display.DLL
PXR800_Display.BAS
Video32.DLL

Initializing and Exiting Libraries

Before calling any library functions, the application must explicitly initialize each library by calling the appropriate `xxx_OpenLibrary()` function. After the application no longer needs access to the libraries, and before the application terminates, the application must call the appropriate `xxx_CloseLibrary()` function.

xxx_CloseLibrary

(for the PXR 800 API library)

`PXRVB_CloseLibrary (ByVal i As Long)`

(for the PXRFRAME library)

`FRAMEVB_CloseLibrary (ByVal i As Long)`

Parameters

The parameter is always 0

Return Value

None

Description

This function will close the specified library, and release any resources that were opened with a corresponding call to the appropriate `xxx_OpenLibrary`.

It is important to no longer access the functions after the library is closed. This will result in an invalid function call, and may result in a General Protection Fault.

xxx_OpenLibrary

(for the PXR 800 API library)

PXRVB_OpenLibrary (ByVal i As Long, ByVal s As Integer) As Boolean

(for the PXRFRAME library)

FRAMEVB_OpenLibrary(ByVal i As Long, ByVal s As Integer) As Boolean

Parameters

The parameters are always (0, 0)

Return Value

If the interface is to a PXR800 API library, the return value is the number of frame grabbers in the system that can be accessed. If the interface is to a PXRFRAME API library, the return value is False on failure, True on success.

Description

The **OpenLibrary** function will open a library to provide access to PXR800 and PXRFrame functions. This library is a DLL. It opens any associated device-driver files.

If the library can not be found, or invalid parameters are provided, a False is returned.

Visual Basic Syntax of the PXR800 API Functions

Following is a list of the PXR800 API functions using their Visual Basic syntax. For detailed information about the functions, please refer to the *PXR800 Frame Grabber User's Guide*.

The Visual Basic support for the PXR800 requires the following header files and libraries

PXR800API.BAS
PXR800.BAS
PXRFRAME.BAS
IMAGINFO.BAS
PXR800.DLL
PXRFRAME.DLL

AllocateFG

PXRVB_AllocateFG (ByVal dwGrabber As Long) As Long

BlockGrabber

PXRVB_BlockGrabber (ByVal fgHandle As Long, ByVal bBlock As Boolean, ByVal pdwDepth As Long, ByVal iwWhen As Long) As Long

CheckError

PXRVB_CheckError (ByVal fgHandle As Long) As Long

FreeFG

PXRVB_FreeFG (ByVal fgHandle As Long)

GetCamera

PXRVB_GetCamera (ByVal fgHandle As Long) As Long

GetContinuousBuffering

PXRVB_GetContinuousBuffering (ByVal fgHandle As Long) As Long

GetDriveMode

PXRVB_GetDriveMode (ByVal fgHandle As Long) As Long

GetFieldLength

PXRVB_GetFieldLength (ByVal fgHandle As Long) As Long

GetFineGain

PXRVB_GetFineGain (ByVal fgHandle As Long) As Long

GetGainRange

PXRVB_GetGainRange (ByVal fgHandle As Long) As Long

GetInterlaceMode

PXRVB_GetInterlaceMode (ByVal fgHandle As Long) As Long

GetLUT

PXRVB_GetLUT (ByVal fgHandle As Long, ByVal dwFirstAddress As Long, ByVal dwLength As Long, ByRef pbLUT As Byte) As Long

GetNyquistFilter

PXRVB_GetNyquistFilter (ByVal fgHandle As Long) As Long

GetOffset

PXRVB_GetOffset (ByVal fgHandle As Long) As Long

GetPixelClock

PXRVB_GetPixelClock (ByVal fgHandle As Long) As Long

GetROI

PXRVB_GetROI (ByVal fgHandle As Long, ByRef pdwROILeft As Long, ByRef pdwROITop As Long, ByRef pdwROIWidth As Long, ByRef pdwROIHeight As Long) As Long

GetRTCStart

PXRVB_GetRTCStart (ByVal fgHandle As Long, ByRef pesEvent As Long, ByRef preRepeat As Long, ByRef psvStart As Long) As Long

GetRTCTiming

PXRVB_GetRTCTiming (ByVal fgHandle As Long, ByRef prtRealTime As RealTime) As Long

GetSyncMode

PXRVB_GetSyncMode (ByVal fgHandle As Long, ByVal icCamera As Long) As Long

GetSyncPolarity

PXRVB_GetSyncPolarity (ByVal fgHandle As Long, ByVal icCamera As Long, ByRef ppoVDrive As Long, ByRef ppoHDrive As Long) As Long

GetTermination

PXRVB_GetTermination (ByVal fgHandle As Long, ByVal icCamera As Long) As Long

GetTime

PXRVB_GetTime (ByVal fgHandle As Long, ByRef pTime As IMG_ImageInfoTiming) As Long

GetTransferMode

PXRVB_GetTransferMode (ByVal fgHandle As Long) As Long

GetTriggerEvent

PXRVB_GetTriggerEvent (ByVal fgHandle As Long, ByRef pdwMask As Long, ByRef pteEvent As Long) As Long

GetVideoFrameSize

PXRVB_GetVideoFrameSize (ByVal fgHandle As Long, ByRef pdwLeftOffset As Long, ByRef pdwTopOffset As Long, ByRef pdwWidth As Long, ByRef pdwHeight As Long) As Long

Grab

PXRVB_Grab (ByVal fgHandle As Long, ByVal pFrame As Long, ByVal dwSkipVSyncs As Long, ByVal geStart As Long, ByVal bSwitchToTrigger As Boolean, ByVal svStart As Long, ByVal flFields As Long, ByVal iwWhen As Long) As Long

HalfHToMics

PXRVB_HalfHToMics (ByVal fgHandle As Long, ByVal dwHalfH As Long) As Long

InvalidateGrabberBuffer

PXRVB_InvalidateGrabberBuffer (ByVal fgHandle As Long, ByVal iwWhen As Long) As Long

IsFinished

PXRVB_IsFinished (ByVal fgHandle As Long, ByVal qiID As Long) As Long

KillQueue

PXRVB_KillQueue (ByVal fgHandle As Long) As Long

MicsToHalfH

PXRVB_MicsToHalfH (ByVal fgHandle As Long, ByVal dwMicroseconds As Long) As Long

ReadIO

PXRVB_ReadIO (ByVal fgHandle As Long) As Long

ReadProtection

PXRVB_ReadProtection (ByVal fgHandle As Long) As Long

ReadRevision

PXRVB_ReadRevision (ByVal fgHandle As Long) As Long

ResetFG

PXRVB_ResetFG (ByVal fgHandle As Long)

ResetRTC

PXRVB_ResetRTC (ByVal fgHandle As Long, ByVal iwWhen As Long) As Long

RTCStart

PXRVB_RTCStart (ByVal fgHandle As Long, ByVal esEvent As Long, ByVal reRepeat As Long, ByVal svImageStart As Long, ByVal iwWhen As Long) As Long

SetCamera

PXRVB_SetCamera (ByVal fgHandle As Long, ByVal icCamera

SetContinuousBuffering

PXRVB_SetContinuousBuffering (ByVal fgHandle As Long, ByVal buContinuous As Long) As Long

SetDriveMode

PXRVB_SetDriveMode (ByVal fgHandle As Long, ByVal dmMode As Long) As Long

SetFineGain

PXRVB_SetFineGain (ByVal fgHandle As Long, ByVal dwGain As Long, ByVal iwWhen As Long) As Long

SetGainRange

PXRVB_SetGainRange (ByVal fgHandle As Long, ByVal dwGain As Long, ByVal iwWhen As Long) As Long

SetInterlaceMode

PXRVB_SetInterlaceMode (ByVal fgHandle As Long, ByVal vfFormat As Long, ByVal iwWhen As Long) As Long

SetLUT

PXRVB_SetLUT (ByVal fgHandle As Long, ByVal dwFirstIndex As Long, ByVal dwLength As Long, ByVal pbLUT As Byte, ByVal iwWhen As Long) As Long

SetNyquistFilter

PXRVB_SetNyquistFilter (ByVal fgHandle As Long, ByVal ifFilter As Long, ByVal iwWhen As Long) As Long

SetOffset

PXRVB_SetOffset (ByVal fgHandle As Long, ByVal IOffset As Long, ByVal iwWhen As Long) As Long

SetPixelClock

PXRVB_SetPixelClock (ByVal fgHandle As Long, ByVal pcClock As Long, ByVal iwWhen As Long) As Long

SetROI

PXRVB_SetROI (ByVal fgHandle As Long, ByVal dwROILeft As Long, ByVal dwROITop As Long, ByVal dwROIWidth As Long, ByVal dwROIHeight As Long, ByVal iwWhen As Long) As Long

SetRTCTiming

PXRVB_SetRTCTiming (ByVal fgHandle As Long, ByRef prtRealTime As RealTime, ByVal dwMask As Long, ByVal iwWhen As Long) As Long

SetSyncMode

PXRVB_SetSyncMode (ByVal fgHandle As Long, ByVal icCamera As Long, ByVal smMode As Long) As Long

SetSyncPolarity

PXRVB_SetSyncPolarity (ByVal fgHandle As Long, ByVal icCamera As Long, ByVal poVDrive As Long, ByVal poHDrive As Long) As Long

SetTermination

PXRVB_SetTermination (ByVal fgHandle As Long, ByVal icCamera As Long, ByVal itTermination As Long) As Long

SetTime

PXRVB_SetTime (ByVal fgHandle As Long, ByRef pTime As IMG_ImageInfoTiming) As Long

SetTransferMode

PXRVB_SetTransferMode (ByVal fgHandle As Long, ByVal tmMode As Long) As Long

SetTriggerEvent

PXRVB_SetTriggerEvent (ByVal fgHandle As Long, ByVal dwTriggerMask As Long, ByVal teEvent As Long) As Long

SetVideoFrameSize

PXRVB_SetVideoFrameSize (ByVal fgHandle As Long, ByVal dwLeftOffset As Long, ByVal dwTopOffset As Long, ByVal dwWidth As Long, ByVal dwHeight As Long, ByVal iwWhen As Long) As Long

TimedWaitFinished

PXRVB_TimedWaitFinished (ByVal fgHandle As Long, ByVal qiID As Long, ByVal dwMs As Long) As Long

Wait

PXRVB_Wait (ByVal fgHandle As Long, ByVal dwSkipVSyncs As Long, ByVal geStart As Long, ByVal bSwitchToTrigger As Boolean, ByVal svStart As Long, ByVal flFields As Long, ByVal iwWhen As Long) As Long

WaitFinished

PXRVB_WaitFinished (ByVal fgHandle As Long, ByVal qiID As Long) As Long

WindowsEventFromQID

PXRVB_WindowsEventFromQID (ByVal fgHandle As Long, ByVal qiID As Long) As Long

WriteIO

PXRVB_WriteIO (ByVal fgHandle As Long, ByVal dwData As Long, ByVal dwMask As Long, ByVal iwWhen As Long) As Long

Visual Basic Syntax of the PXRFrame API Functions

Following is a list of the PXRFrame API functions using their Visual Basic syntax. For detailed information about the functions, please refer to the *PXR800 Frame Grabber User's Guide*.

The Visual Basic support for the PXR frame library requires the following header files and libraries

PXRFRAME.BAS
IMAGINFO.BAS
PXRFRAME.DLL

AllocateBuffer

FRAMEVB_AllocateBuffer (ByVal dwWidth As Long, ByVal dwPitch As Long, ByVal dwHeight As Long, ByVal ptType As Long, ByVal blmImageInfo As Boolean) As Long

CopyFrame

FRAMEVB_CopyFrame (ByVal pSourceFrame As Long, ByVal dwSourceX As Long, ByVal dwSourceY As Long, ByVal pDestFrame As Long, ByVal dwDestX As Long, ByVal dwDestY As Long, ByVal dwWidth As Long, ByVal dwHeight As Long) As Long

FrameBits

FRAMEVB_FrameBits (ByVal pFrame As Long) As Long

FrameBuffer

FRAMEVB_not available

FrameFromPointer

not available

FrameHeight

FRAMEVB_FrameHeight (ByVal pFrame As Long) As Long

FramePitch

FRAMEVB_FramePitch (ByVal pFrame As Long) As Long

FrameType

FRAMEVB_FrameType (ByVal pFrame As Long) As Long

FrameWidth

FRAMEVB_FrameWidth (ByVal pFrame As Long) As Long

FreeFrame

FRAMEVB_FreeFrame (ByVal pFrame As Long)

GetColumn

FRAMEVB_GetColumn (ByVal pFrame As Long, ByRef pvBuffer As Any, ByVal dwColumn As Long) As Long

GetPixel

FRAMEVB_GetPixel (ByVal pFrame As Long, ByRef pvBuffer As Any, ByVal dwColumn As Long, ByVal dwRow As Long) As Long

GetRectangle

FRAMEVB_GetRectangle (ByVal pFrame As Long, ByRef pvBuffer As Any, ByVal dwColumn As Long, ByVal dwRow As Long, ByVal dwWidth As Long, ByVal dwHeight As Long) As Long

GetRow

FRAMEVB_GetRow (ByVal pFrame As Long, ByRef Buf As Any, ByVal dwRow As Long) As Long

GetTimeStamp

FRAMEVB_GetTimeStamp (ByVal pFrame As Long, ByRef pTimeStamp As IMG_ImageInfo) As Long

PutColumn

FRAMEVB_PutColumn (ByRef pvBuf As Any, ByVal pFrame As Long, ByVal dwColumn As Long) As Long

PutPixel

FRAMEVB_PutPixel (ByRef pvPixel As Any, ByVal pFrame As Long, ByVal dwColumn As Long, ByVal dwRow As Long) As Long

PutRectangle

FRAMEVB_PutRectangle (ByRef pvBuf As Any, ByVal pFrame As Long, ByVal dwColumn As Long, ByVal dwRow As Long, ByVal dwWidth As Long, ByVal dwHeight As Long) As Long

PutRow

FRAMEVB_PutRow (ByRef Buf As Any, ByVal pFrame As Any, ByVal dwRow As Long) As Long

ReadBin

FRAMEVB_ReadBin (ByVal pFrame As Long, ByVal szFilename As String) As Long

ReadBMP

FRAMEVB_ReadBMP (ByVal pFrame As Long, ByVal szFilename As String) As Long

ReadPNG

FRAMEVB_ReadPNG (ByVal pFrame As Long, ByVal szFilename As String)
As Long

WriteBin

FRAMEVB_WriteBin (ByVal pFrame As Long, ByVal szFilename As String,
ByVal bOverwrite As Boolean) As Long

WriteBMP

FRAMEVB_WriteBMP (ByVal pFrame As Long, ByVal szFilename As String,
ByVal bOverwrite As Boolean) As Long

WritePNG

FRAMEVB_WritePNG (ByVal pFrame As Long, ByVal szFilename As String,
ByVal bOverwrite As Boolean) As Long

Visual Basic Syntax of the Video Display Functions

Following is a list of the video display functions. For detailed information about the functions, please refer to the section [*Displaying Video in Visual Basic Applications*](#)~~*Displaying Video in Visual Basic Applications*~~

The Visual Basic support for video display functions requires the following header files and libraries

PXR800_DISPLAY.DLL
PXR800_DISPLAY.BAS
VIDEO_32.DLL

IMG_VB_PaintDisplayWindow

IMG_VB_PaintDisplayWindow (ByVal hDC As Long, ByVal PXR_Frame As Long, ByVal x As Long, ByVal y As Long, ByVal dx As Long, ByVal dy As Long)

IMG_VB_SizeDisplayWindow

IMG_VB_SizeDisplayWindow (ByVal x As Long, ByVal y As Long, ByVal dx As Long, ByVal dy As Long)