
Pegasus Development Kit

Programmer's Reference

Version.1.00

Pegasus Development Kit is out of a support.

(C) 2007 Thomson.All rights reserved.
Grass Valley is a trademark of Thomson.





Table Of Contents

TABLE OF CONTENTS	2
PRELIMINARIES	3
Run The Sample Application	3
Development Using Visual C++ 2005	3
Compile.....	3
Link.....	3
Application Development in other environments.....	4
Sample Programs.....	4
Source Code	4
API REFERENCE	5
Initialization and Release	5
Input Selection	6
The Preview	8
The Capture	10
The Hardware Settings	12
The Input Settings.....	14
Other functions	15
VARIABLES AND CONSTANTS REFERENCE	17
Return values of APIs	17
PCEPreviewType.....	17
PCEPreviewPeriod	18
PCEInputChannel.....	18
PCEHardwareParam	18
PCEInputParam	19
PCECaptureEvent	21
PCEParamType	21
PCEFieldOrder.....	21
PCEFRACTION	21
PCEFORMAT	22

Preliminaries

Run The Sample Application

1. Install HDRECS driver.

* If you have installed HQ RECORDER, you don't have to do this operation.

Run Drivers\setup.exe in the HDRECS setup image.

2. Install Runtimes.

* If you have installed HQ RECORDER, you don't have to do this operation.

Run the following files.

- runtime\vc redistrib_x86.exe
Visual C++ 2005 Redistributable Package (x86)
- runtime\DirectX\DXSETUP.exe

3. Run The Sample Application.

Put the following files in the same directory.

- sample\Release\sample.exe
- bin*

Run sample.exe.

Development Using Visual C++ 2005

Compile

When compiling include the file below.

inc\pceapi.h.

Link

When linking include the file below.

lib\pceapi.lib.



Application Development in other environments

This Development Kit was created using Microsoft *Visual C++ 2005*. It can be used in Microsoft Visual C++ 2005 compatible environments.

We can not be sure of the results if other development environments are used.

If using another development environment, please consult the manufacturer for usage instructions with this development kit.

Sample Programs

There are some sample programs included in the development kit for reference. The sample programs were made in *Visual C++ 2005*. The included sample programs are listed in **sample\readme.txt**.

Source Code

The source code included with the sample programs can be used or modified for incorporation into another application. However, there will be no support from Grass Valley for such software.

Grass Valley grants permission to use and modify the included source code. Distribution in any form of the included source code is prohibited. (Objects or compiled applications can be distributed, but any source code material that contains source code included with this SDK may not be distributed.)

Grass Valley cannot be held responsible for any results of applications developed using the development kit or its included source code.



API Reference

Initialization and Release

PCEAPI PCE_Initialize(PCEHANDLE *IpHandle, DWORD reserved)

This function initializes the SDK. Applications are required to use PCE_Initialize before they use any other SDK's functions.

[Parameters]

IpHandle

[out] Pointer to a variable that receives a handle to the SDK.

reserved

[in] Reserved; must be zero.

PCEAPI PCE_Release(PCEHANDLE handle)

This function releases the SDK.

[Parameters]

handle

[in] Handle to the SDK.

Input Selection

PCEAPI PCE_SelectInput(PCEHANDLE handle, PCEInputChannel channel, DWORD dwFormatIndex)

This function selects an input.

[Parameters]

handle

[in] Handle to the SDK.

channel

[in] Specifies the input channel. If this parameter is PCEInputChannel_NOCHANNEL, no input is selected.

dwFormatIndex

[in] Specifies the format index.

PCEAPI PCE_GetInput(PCEHANDLE handle, PCEInputChannel *lpChannel, LPDWORD lpdwFormatIndex)

This function retrieves the current input.

[Parameters]

handle

[in] Handle to the SDK.

lpChannel

[out] Pointer to a variable that retrieves the current input channel. If no input has been selected, this variable retrieves PCEInputChannel_NOCHANNEL. If this parameter is NULL, the input channel isn't retrieved.

lpdwFormatIndex

[out] Pointer to a variable that retrieves the current format index. If no input has been selected, this variable retrieves -1(0xffffffff). If this parameter is NULL, the format index isn't retrieved.

PCEAPI PCE_GetInputFormatNum(PCEHANDLE handle, PCEInputChannel channel, LPDWORD lpdwFormatNum)

This function retrieves the number of formats in the specified input channel.

[Parameters]

handle

[in] Handle to the SDK.

channel

[in] Specifies the input channel.

lpdwFormatNum

[out] Pointer to a unsigned integer that retrieves the number of formats.

PCEAPI PCE_GetInputFormat(PCEHANDLE handle, PCEInputChannel channel, DWORD dwFormatIndex, PCEFORMAT *lpFormat)

This function retrieves the format description.

[Parameters]

handle

[in] Handle to the SDK.

channel

[in] Specifies the input channel.

dwFormatIndex

[in] Specifies the format index.

lpFormat

[out] Pointer to a variable that retrieves the format description.

The Preview

PCEAPI PCE_StartPreview(PCEHANDLE handle, HWND hWnd, LONG nX, LONG nY, LONG nWidth, LONG nHeight)

This function starts the preview.

[Parameters]

handle

[in] Handle to the SDK.

hWnd

[in] Specifies the handle to the window.

nX

[in] Specifies the x-coordinate of the overlay's upper-left corner of the window's client area.

nY

[in] Specifies the y-coordinate of the overlay's upper-left corner of the window's client area.

nWidth

[in] Specifies the width of the window.

nHeight

[in] Specifies the height of the window.

[Remarks]

PCE_StartPreview creates the child window of the specified window and displays the overlay in the child window.

PCEAPI PCE_StopPreview(PCEHANDLE handle)

This function stops the preview.

[Parameters]

handle

[in] Handle to the SDK.



PCEAPI PCE_UpdatePreview(PCEHANDLE handle, LONG nX, LONG nY, LONG nWidth, LONG nHeight)

This function updates the overlay's position and size.

[Parameters]

handle

[in] Handle to the SDK.

nX

[in] Specifies the x-coordinate of the overlay's upper-left corner of the window's client area.

nY

[in] Specifies the y-coordinate of the overlay's upper-left corner of the window's client area.

nWidth

[in] Specifies the width of the window.

nHeight

[in] Specifies the height of the window.

The Capture

PCEAPI PCE_StartCapture(PCEHANDLE handle, LPCWSTR lpwszFileName, BOOL bOverwrite, DWORD dwFlag, PCECaptureCallback pfnCapture, LPVOID lpArg)

This function starts the capture.

[Parameters]

handle

[in] Handle to the SDK.

lpwszFileName

[in] Specifies the file name. Note that this parameter is the unicode string.

bOverwrite

[in] Specifies whether to overwrite the file, if it already exists. If this parameter is TRUE, an overwrite occurs. Otherwise, an overwrite doesn't occur.

dwFlag

[in] Specifies the flag. This parameter can be a combination of the following flags.

PCE_CAPTUREFLAG_FORCEAUDIO_16BIT_2CH

Capture the audio with 16bit/2ch.

pfnCapture

[in] Specifies the callback function that retrieves the capture event.

lpArg

[in] Pointer to a variable to be passed to the callback function.

[Remarks]

The callback function for the capture is the following format.

**void (*PCECaptureCallback)(LPVOID lpArg, PCECaptureEvent captureEvent, DWORD dwParam)
lpArg equals lpArg of PCE_StartCapture.**

**When the capture started actually, the callback function receives PCECaptureEvent_Start.
(dwParam is zero)**

**When the capture stopped actually, the callback function receives PCECaptureEvent_Stop,
(dwParam is zero)**

**When the error occurred, the callback function receives PCECaptureEvent_Error.
(dwParam is the error cause)**



Pegasus Development Kit

PCEAPI PCE_StopCapture(PCEHANDLE handle)

This function stops the capture.

[Parameters]

handle

[in] Handle to the SDK

[Remarks]

Even if the callback function receives PCECaptureEvent_Error, Applications must call PCE_StopCapture.

The Hardware Settings

PCEAPI PCE_SetHardwareParam(PCEHANDLE handle, PCEHardwareParam param, DWORD dwValue)

This function changes the hardware parameter. But the hardware parameter isn't actually updated until PCE_UpdateHardwareParam is called.

[Parameters]

handle

[in] Handle to the SDK.

param

[in] Specifies the hardware parameter's name.

dwValue

[in] Specifies the replacement value.

PCEAPI PCE_UpdateHardwareParam(PCEHANDLE handle)

This function updates the hardware parameters that were changed by using PCE_SetHardwareParam.

[Parameters]

handle

[in] Handle to the SDK.

[Remarks]

If the preview is running, PCE_UpdateHardwareParam re-selects the current input. Therefore the preview stops for a moment.



Pegasus Development Kit

PCEAPI PCE_GetHardwareParam(PCEHANDLE handle, PCEHardwareParam param, LPDWORD lpdwValue, PCEParamType type);

This function retrieves the hardware parameter.

[Parameters]

handle

[in] Handle to the SDK.

param

[in] Specifies the hardware parameter's name.

lpdwValue

[out] Pointer to a unsigned integer that receives the value.

type

[in] Specifies the type of the hardware parameter.

The Input Settings

PCEAPI PCE_SetInputParam(PCEHANDLE handle, PCEInputChannel channel, PCEInputParam param, DWORD dwValue)

This function changes the input parameter of the specified input channel. The input parameter is instantly updated.

[Parameters]

handle

[in] Handle to the SDK.

channel

[in] Specifies the input channel.

param

[in] Specifies the input parameter's name.

dwValue

[in] Specifies the replacement value.

PCEAPI PCE_GetInputParam(PCEHANDLE handle, PCEInputChannel channel, PCEInputParam param, LPDWORD lpdwValue, PCEParamType type)

This function retrieves the input parameter of the specified input channel.

[Parameters]

handle

[in] Handle to the SDK.

channel

[in] Specifies the input channel.

param

[in] Specifies the input parameter's name.

lpdwValue

[out] Pointer to a unsigned integer that receives the value.

type

[in] Specifies the type of the input parameter.



Other functions

PCEAPI PCE_GetHardwareStatus(PCEHANDLE handle, PCEHardwareStatus status, LPDWORD lpdwValue)

This function retrieves the hardware status.

[Parameters]

handle

[in] Handle to the SDK.

status

[in] Specifies the hardware status.

lpdwValue

[out] Pointer to a unsigned integer that receives the value.

PCEAPI PCE_SetPreviewSettings(PCEHANDLE handle, PCEPreviewType type, PCEPreviewPeriod period)

This function changes the preview settings. To enable the settings, applications are required to re-initialize the SDK.

[Parameters]

handle

[in] Handle to the SDK.

type

[in] Specifies the view type of the overlay.

period

[in] Specifies the update period of the overlay.



Pegasus Development Kit

PCEAPI PCE_GetPreviewSettings(PCEHANDLE handle, PCEPreviewType *IpType, PCEPreviewPeriod *IpPeriod)

This function retrieves the current preview settings.

[Parameters]

handle

[in] Handle to the SDK.

IpType

[out] Pointer to a variable that receives the view type of the overlay. If this parameter is NULL, the view type isn't retrieved.

IpPeriod

[out] Pointer to a variable that receives the update period of the overlay. If this parameter is NULL, the update period isn't retrieved.

Variables and Constants Reference

Return values of APIs

PCESUCCEEDED macro and PCEFAILED macro can check whether the API succeeded.

PCE_S_OK	The function succeeded.
PCE_S_FALSE	The function succeeded. But nothing has been done.
PCE_E_FAILED	The function failed.
PCE_E_NOMEMORY	Enough memory isn't available.
PCE_E_INVALIDPARAMETER	The parameter is invalid.
PCE_E_NOTIMPLEMENTED	The function has not been implemented.
PCE_E_INTERNAL	The internal error occurred.
PCE_E EDIUSEXIST	EDIUS has been running.
PCE_E_NOTINITIALIZED	The SDK has not been initialized.
PCE_E_INVALIDSTATE	The current state is invalid to run the function.
PCE_E_DISKFULL	The storage is full.
PCE_E_TOOSLOW	The data transfer rate is too slow.
PCE_E_CANNOTOPEN	The file cannot be opened.
PCE_E_NOSTREAM	No stream was received.
PCE_E_NODEVICE	No device was found.
PCE_E_INVALIDMODULE	The DLL module is invalid.
PCE_E_NOTSUPPORTED	The function has not been supported.
PCE_E_FILEEXIST	The file already exists.

PCEPreviewType

This variable indicates the view type of the overlay.

PCEPreviewType_Default	The current setting is used to display the overlay.
PCEPreviewType_DDraw	DirectDraw is used to display the overlay.
PCEPreviewType_D3D	Direct3D is used to display the overlay.

PCEPreviewPeriod

This variable indicates the update period of the overlay.

PCEPreviewPeriod_Default	The current setting is used to update the overlay.
PCEPreviewPeriod_Field	The overlay is updated every field.
PCEPreviewPeriod_Frame	The overlay is updated every frame.

PCEInputChannel

This variable indicates the input channel.

PCEInputChannel_NOCHANNEL	No channel.
PCEInputChannel_Composite	Composite.
PCEInputChannel_S	S-Video.
PCEInputChannel_Component	Component.
PCEInputChannel_HDMI	HDMI.
PCEInputChannel_DVI	DVI.
PCEInputChannel_RGB	RGB.

PCEHardwareParam

This variable indicates the hardware parameter's name.

PCEHardwareParam_HQParamQ	This parameter indicates video quality of the HQ encoding. The minimum value is the highest quality.
PCEHardwareParam_HQParamSize	This parameter indicates the maximum bitrate. The value 100 percent means the same bitrate as an uncompressed video.
PCEHardwareParam_Ignore422Filter	This parameter indicates whether to ignore YUV422 filter. PCE_HARDWAREPARAM_IGNORE422FILTER_OFF This value means to use YUV422 filter. PCE_HARDWAREPARAM_IGNORE422FILTER_ON This value means to ignore YUV422 filter.

PCEInputParam

This variable indicates the input parameter's name.

The following parameters are available when the input channel is Composite, S-Video or Component.

PCEInputParam_Brightness	Brightness of the video.
PCEInputParam_Contrast	Contrast of the video.
PCEInputParam_Hue	Hue of the video. This parameter isn't available at 1080i and 720p.
PCEInputParam_Saturation	Saturation of the video.
PCEInputParam_Sharpness	Sharpness of the video. This parameter isn't available at 1080i and 720p.
PCEInputParam_AGCSwitch	Whether to use the auto gain control. PCE_INPUTPARAM_AGCSWITCH_OFF Disable the auto gain control. PCE_INPUTPARAM_AGCSWITCH_ON Enable the auto gain control.
PCEInputParam_AGCGain	This parameter indicates the gain value when PCEInputParam_AGCSwitch is PCE_INPUTPARAM_AGCSWITCH_OFF.
PCEInputParam_SetupLevel	NTSC setup level. This parameter is available except Component. PCE_INPUTPARAM_SETUPLEVEL_0IRE 0IRE (Japan) PCE_INPUTPARAM_SETUPLEVEL_75IRE 7.5IRE (US)
PCEInputParam_3DYC	Whether to enable 3D Y/C separation at NTSC. This parameter is available only when the input channel is Composite. PCE_INPUTPARAM_3DYC_OFF Disable 3D Y/C separation. PCE_INPUTPARAM_3DYC_ON Enable 3D Y/C separation.

PCEInputParam_3DNR	Level of 3D noise reduction. This parameter isn't available at 1080i and 720p.
PCE_INPUTPARAM_3DNR_OFF	Disable 3D noise reduction.
PCE_INPUTPARAM_3DNR_WEAK	The level of 3D noise reduction is weak.
PCE_INPUTPARAM_3DNR_MEDIUM	The level of 3D noise reduction is medium.
PCE_INPUTPARAM_3DNR_STRONG	The level of 3D noise reduction is strong.

The following parameters is available when the input channel is RGB.

PCEInputParam_RGBGain	Gain value.
PCEInputParam_Screen1H	Horizontal position of the 1280x1024 screen. The default value is 50. If the value is smaller than 50, the screen moves to left. If the value is larger than 50, the screen moves to right.
PCEInputParam_Screen1V	Vertical position of the 1280x1024 screen. The default value is 50. If the value is smaller than 50, the screen moves to up. If the value is larger than 50, the screen moves to down.
PCEInputParam_Screen2H	Horizontal position of the 1024x768 screen. The default value is 50. If the value is smaller than 50, the screen moves to left. If the value is larger than 50, the screen moves to right.
PCEInputParam_Screen2V	Vertical position of the 1024x768 screen. The default value is 50. If the value is smaller than 50, the screen moves to up. If the value is larger than 50, the screen moves to down.
PCEInputParam_Screen3H	Horizontal position of the 800x600 screen. The default value is 50. If the value is smaller than 50, the screen moves to left. If the value is larger than 50, the screen moves to right.
PCEInputParam_Screen3V	Vertical position of the 800x600 screen. The default value is 50. If the value is smaller than 50, the screen moves to up. If the value is larger than 50, the screen moves to down.
PCEInputParam_Screen4H	Horizontal position of the 640x480 screen. The default value is 50. If the value is smaller than 50, the screen moves to left. If the value is larger than 50, the screen moves to right.
PCEInputParam_Screen4V	Vertical position of the 640x480 screen. The default value is 50. If the value is smaller than 50, the screen moves to up. If the value is larger than 50, the screen moves to down.

PCECaptureEvent

This variable indicates the capture event that PCECaptureCallback receives.

PCECaptureEvent_Start	When the capture started actually, the callback function receives this event.
PCECaptureEvent_Stop	When the capture stopped actually, the callback function receives this event.
PCECaptureEvent_Error	When the capture error occurred, the callback function receives this event.

PCEParamType

This variable indicates type of value that is received at PCE_GetHardwareParam or PCE_GetInputParam.

PCEParamType_Current	Current value.
PCEParamType_Default	Default value.
PCEParamType_Max	Maximum value.
PCEParamType_Min	Minimum value.

PCEFieldOrder

This variable indicates the field order of the input format.

PCEFieldOrder_Progressive	Progressive.
PCEFieldOrder_TopFieldFirst	Top field first.
PCEFieldOrder_BottomFieldFirst	Bottom field first.

PCEFRACTION

This struct indicates fraction.

nNumerator	Numerator
nDenominator	Denominator



PCEFORMAT

This struct indicates the format description.

wszName	Format name. The maximum length of string is PCE_MAX_FORMAT_STRING. (includes \0)
nWidth	Video width.
nHeight	Video height.
Aspect	Video aspect ratio.
frameRate	Video frame rate.
fieldOrder	Video field order.