

roduction

First topic in project 'ImperX Inc. VCE ANCB FrameGrabber SDK'

VCEANCB Functions Reference

Initialize/Destroy functions

VCEANCB_Init	Initialize VCEANCB
VCEANCB_Done	Close VCEANCB

Video Source controls

VCEANCB_setBrightness	Set brightness level.
VCEANCB_setContrast.htm	Set contrast level.
VCEANCB_setSaturation	Set saturation level.
VCEANCB_setHue	Set hue level.
VCEANCB_setGamma	Set gamma correction level.
VCEANCB_setInputChannel	Select input channel.
VCEANCB_getVideoStandard	Get current video standard (PAL, SECAM or NTSC).
VCEANCB_SetColorKiller	Set level of chrominance subcarrier to detect color.
VCEANCB_IsVideoSource	Returns status of video source connection.

Grabbing functions

VCEANCB_Prepare	Set options to VCEANCB before frame snapping.
VCEANCB_PrepareGrab	Set options to VCEANCB and prepare to stream-grabbing.
VCEANCB_SnapFrame	Snap one frame
VCEANCB_StartGrab	Start grabbing stream
VCEANCB_StopGrab	Stop grabbing stream
VCEANCB_GrabFrame_Callback	User-defined callback function
VCEANCB_LockBuffer	Get buffer with snapped data
VCEANCB_UnlockBuffer	Release buffer

B Management functions

VCEANCB_GetDIB	Generate DIB from snapped frame
VCEANCB_CopyDIB	Duplicate DIB.
VCEANCB_ConvertPixels2DIB	Generate DIB from raw bytes recieved from digitizer
VCEANCB_ReleaseDIB	Delete DIB
VCEANCB_DrawDIB	Draw DIB to DeviceContext
VCEANCB_SaveDIBToFile	Save DIB to BMP file

igger functions

VCEANCB_GetTriggerStatus	Get status of external trigger
--	--------------------------------

ror lookup functions

VCEANCB_CardLastError	Get last VCEANCB error
VCEANCB_SystemLastError	Get last system error

VCEANCB_CardLastError

The **VCEANCB_CardLastError** function returns value of last VCEANCB function call.

```
VCEANCB_Error VCEANCB_CardLastError ();
```

turn values:

Returns value of last VCEANCB function. Usefull in conjunction with **VCEANCB_Init()** function.

VCEANCB_DefaultConfig

The **VCEANCB_DefaultConfig** function fills VCEANCB_CameraConfig structure with default configuration data.

```
VCEANCB_Error VCEANCB_DefaultConfig (  
    HANDLE hVCEANCB, // Handle to card  
    VCEANCB_CameraConfig* pCameraConfig // Data to DefaultConfig  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

pCameraConfig

[in] Pointer to a [VCEANCB_CameraConfig](#) structure.

See also:

[VCEANCB_CameraConfig](#)

VCEANCB_Done

The **VCEANCB_Done** function deinitializes FrameLink card.

```
VCEANCB_Error VCEANCB_Done (  
    HANDLE hVCEANCB // Handle to card  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle is bad.
---	-------------------------

See also:

[VCEANCB_Init](#)

VCEANCB_DrawDIB

The **VCEANCB_DrawDIB** function draws Device Independent Bitmap(DIB) generated by [VCEANCB_GetDIB](#) to device context (DC).

```
VCEANCB_Error VCEANCB_DrawDIB (  
    HDC hDC, // Handle to DC  
    HGLOBAL hDIB, // Handle to DIB  
    WORD xPos, // X position  
    WORD yPos, // Y position  
    WORD xSize, // Width of destination image  
    WORD ySize // Height of destination image  
);
```

Parameter:

hDC

[in] Handle to a device context.

hDIB

[in] Pointer to DIB handle.

xPos

[in] Specifies the x-coordinate, in logical units, of the upper-left corner of the destination rectangle.

yPos

[in] Specifies the y-coordinate, in logical units, of the upper-left corner of the destination rectangle.

xSize

[in] Specifies the width, in logical units, of the destination rectangle.

ySize

[in] Specifies the height, in logical units, of the destination rectangle.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hDIB or hDC handle is bad (possibly NULL).
VCEANCB_Err_noMemory	Not enough memory.

[VCEANCB_Err_UnknownError](#)

Unknown (possibly system) error occurred. To get extended error information, call [VCEANCB_SystemLastError](#).

marks

This function sets DIBits from DIB to DC. If xSize and/or ySize is zero function uses original width and/or height of image. If xSize and/or ySize not equal to origin sizes of image, function uses **StretchDIBits** Microsoft® Windows® GDI function, otherwise function uses **SetDIBitsToDevice** function which is faster.

see also:

[VCEANCB_SnapFrame](#), [VCEANCB_ReleaseDIB](#), [VCEANCB_GetDIB](#), [VCEANCB_SaveDIBToFile](#)

VCEANCB_GetDIB

The **VCEANCB_GetDIB** function generates Device Independent Bitmap (DIB) from snapped frame.

```
VCEANCB_Error VCEANCB_GetDIB (  
    HANDLE hVCEANCB, // Handle to card  
    HGLOBAL* phDIB // Pointer to DIB  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

phDIB

[out] Pointer to DIB handle.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	<i>hVCEANCB</i> handle or <i>phDIB</i> pointer is bad (possibly NULL).
VCEANCB_Err_notGrabbed	VCEANCB_SnapFrame was not called.
VCEANCB_Err_bufferBusy	Snapping buffers are busy.
VCEANCB_Err_notSupported	Unsupported camera operation mode.
VCEANCB_Err_noMemory	Not enough memory.

Remarks:

VCEANCB_GetDIB function generates Device Independent Bitmap explained in **Device-Independent Bitmaps** section of *Windows® GDI*. To free memory used by DIB call [VCEANCB_ReleaseDIB](#).

See also:

[VCEANCB_SnapFrame](#), [VCEANCB_ReleaseDIB](#),
[VCEANCB_DrawDIB](#), [VCEANCB_SaveDIBToFile](#)

VCEANCB_getVideoStandard

The **VCEANCB_getVideoStandard** checks current video standard.

```
VCEANCB_Error VCEANCB_getVideoStandard (  
    HANDLE hVCEANCB, // Handle to card  
    VCEANCB_VideoStandard* pVideoStandard, // pointer to receive video  
    VCEANCB_VideoFreq* pVideoFrequency // pointer to receive video fre  
  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

pVideoStandard

[out] Current video standard. Could be NULL.

pVideoFrequency

[out] Current video frequency. Could be NULL.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle is bad (possibly NULL).
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Remarks:

This function checks for current video standard and video frequency.

See also:

[VCEANCB_Prepare](#), [VCEANCB_CameraConfig](#),
[VCEANCB_VideoStandard](#) [VCEANCB_VideoFreq](#)

VCEANCB_Init

The **VCEANCB_Init** function initializes FrameGrabber.

```
HANDLE VCEANCB_Init();
```

Return Values:

If the function succeeds, the return value is handle to card.

If the function fails, the return value is NULL.

To get extended error information, call [VCEANCB_CardLastError](#)

The error value can be one of the following values.

VCEANCB_Err_noMemory	Not enough memory.
VCEANCB_Err_noDriver	Driver is not installed.
VCEANCB_Err_noDevicePresent	No VCEANCB device present.
VCEANCB_Err_UnknownError	Unknown (possibly system) error. To get extended error information, call VCEANCB_SystemLastError .

See also:

[VCEANCB_Done](#)

VCEANCB_LockBuffer

[This is preliminary documentation and subject to change.]

The **VCEANCB_LockBuffer** function gets raw bytes from snapped frame.

```
VCEANCB_Error VCEANCB_LockBuffer (  
    HANDLE hVCEANCB, // Handle to card  
    BYTE** ppBuffer, // Pointer to pointer to buffer  
    DWORD* pdwSize // Pointer to size of buffer  
);
```

parameter:

hVCEANCB

[in] Handle to FrameLink card.

ppBuffer

[out] Pointer to pointer to buffer.

pdwSize

[out] Pointer to size of buffer

return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or ppBuffer pointer is bad (possibly NULL).
VCEANCB_Err_notGrabbed	VCEANCB_SnapFrame was not called.
VCEANCB_Err_bufferBusy	Snapping buffers are busy.

marks:

The buffer is a raw bytes in YUV 4:2:2 format received from digitizer for color image(see table bellow) or "Y only" format, where each byte represent Luma component of YUV data.

Cb0	Y0	Cr0	Y1	Cb2	Y2	Cr2	Y3
-----	----	-----	----	-----	----	-----	----

Even field placed at begin of returned buffer. Start of odd field could be

found using next formula:

```
if(monochrome) { // For monochrome image
    sourceWidth = dwWidth;
    frameShift = dwWidth*(dwHeight/2)/512;
    if((dwWidth*(dwHeight/2))%512)
        frameShift++;
    frameShift*=512;
}
else { // For color image
    sourceWidth = 2*dwWidth;
    frameShift = dwWidth*dwHeight/512;
    if((dwWidth*dwHeight)%512)
        frameShift++;
    frameShift*=512;
}
```

Where

monochrome

Monochrome flag. Used in VCEANCB_SnapData structure (corresponding PAL or NTSC field of VCEANCB_CameraConfig structure)

dwWidth

Width of image in pixels

dwHeight

Height of image in pixels

frameShift

Shift of odd field relative to start of buffer in bytes

sourceWidth

Width of line in source buffer in bytes

e also:

[VCEANCB_SnapFrame](#), [VCEANCB_UnlockBuffer](#)

VCEANCB_Prepare

The **VCEANCB_Prepare** function prepares FrameLink card to snap one frame from camera.

```
VCEANCB_Error VCEANCB_Prepare (  
    HANDLE hVCEANCB, // Handle to card  
    VCEANCB_CameraConfig* pCameraConfig // Data to prepare  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

pCameraConfig

[in] Pointer to a [VCEANCB_CameraConfig](#) structure.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_notInitialized	FrameLink has not been successfully initialized.
VCEANCB_Err_bufferBusy	One of snapping buffers is busy.
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Remarks:

This function deletes all snapping buffer and sends configuration command to FrameLink card. **Note:** Use this function **only** if you would like to use **VCEANCB_SnapFrame()** function.

If you would like to use stream grabbing function **VCEANCB_StartGrab** you need to call **VCEANCB_PrepareGrab()** function.

See also:

[VCEANCB_CameraConfig](#), [VCEANCB_SnapFrame](#),

VCEANCB PrepareGrab

VCEANCB_PrepareGrab

The **VCEANCB_PrepareGrab** function prepares FrameLink card to grab stream from camera.

```
VCEANCB_Error VCEANCB_PrepareGrab (  
    HANDLE hVCEANCB, // Handle to card  
    VCEANCB_CameraConfig* pCameraConfig // Data to prepare  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

pCameraConfig

[in] Pointer to a [VCEANCB_CameraConfig](#) structure.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_notInitialized	FrameLink has not been successfully initialized.
VCEANCB_Err_bufferBusy	One of snapping buffers is busy.
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Remarks:

This function deletes all snapping buffer and sends configuration command to FrameLink card. **Note:** Use this function **only** if you would like to use **VCEANCB_StartGrab()** function.

If you would like to use one frame snapping function

VCEANCB_SnapFrame you need to call **VCEANCB_Prepare()** function.

See also:

[VCEANCB_CameraConfig](#), [VCEANCB_StartGrab](#),

VCEANCB Prepare

VCEANCB_ReleaseDIB

The **VCEANCB_ReleaseDIB** function deletes DIB generated by [VCEANCB_GetDIB](#).

```
VCEANCB_Error VCEANCB_ReleaseDIB (  
    HANDLE hVCEANCB, // Handle to card  
    HGLOBAL* phDIB // Pointer to DIB  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

phDIB

[in/out] Pointer to DIB handle.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#)

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	phDIB pointer or handle this pointer point to is bad (Possibly NULL).
---	---

See also:

[VCEANCB_GetDIB](#)

VCEANCB_SaveDIBToFile

The **VCEANCB_SaveDIBToFile** function saves Device Independent Bitmap (DIB) to Bitmap (BMP) file.

```
VCEANCB_Error VCEANCB_SaveDIBToFile (  
    HGLOBAL hDIB, // Handle to DIB  
    char* filename // Filename to save the file  
);
```

Parameter:

hDIB

[in] Handle to DIB.

filename

[in] Path to file to save image.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hDIB handle or filename is bad (Possibly NULL).
VCEANCB_Err_noMemory	Not enough memory.
VCEANCB_Err_UnknownError	Unknown (possibly system) error. To get extended error information, call VCEANCB_SystemLastError .

Remarks:

See also:

[VCEANCB_SnapFrame](#), [VCEANCB_GetDIB](#), [VCEANCB_ReleaseDIB](#), [VCEANCB_DrawDIB](#)

VCEANCB_setBrightness

The **VCEANCB_setBrightness** sets brightness adjustment level.

```
VCEANCB_Error VCEANCB_setBrightness (  
    HANDLE hVCEANCB, // Handle to card  
    BYTE brightness // Brightness level  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

brightness

[in] Brightness level

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Constants:

This function sets brightness adjustment level.

Constant	Value	Description
VCEANCB_BrightnessMax	256	Maximum brightness level
VCEANCB_BrightnessMin	0	Minimum brightness level
VCEANCB_BrightnessDef	128	ITU brightness level

See also:

[VCEANCB Prepare](#), [VCEANCB CameraConfig](#)

VCEANCB_setContrast

The `VCEANCB_setContrast` sets contrast adjustment level.

```
VCEANCB_Error VCEANCB_setContrast (  
    HANDLE hVCEANCB, // Handle to card  
    signed char contrast // Contrast level  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

contrast

[in] Contrast level

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Constants:

This function sets contrast adjustment level.

Constant	Value	Description
VCEANCB_ContrastMax	127	Maximum contrast level (equal to 1.984)
VCEANCB_ContrastMin	0	Minimum contrast level (equal to 0.0)
VCEANCB_ContrastDef	68	ITU contrast level (equal to 1.063)

See also:

[VCEANCB_Prepare](#), [VCEANCB_CameraConfig](#)

VCEANCB_setGamma

The `VCEANCB_setGamma` sets gamma correction.

```
VCEANCB_Error VCEANCB_setGamma (  
    HANDLE hVCEANCB, // Handle to card  
    float gamma // Gamma correction level  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

gamma

[in] Gamma correction level

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Remarks:

This function rebuilds lookup table with new gamma correction.

Constant	Value	Description
VCEANCB_SaturationDef	1.0f	Normal gamma correction.

See also:

[VCEANCB_Prepare](#), [VCEANCB_CameraConfig](#)

VCEANCB_setHue

The **VCEANCB_setHue** sets hue adjustment level.

```
VCEANCB_Error VCEANCB_setHue (  
    HANDLE hVCEANCB, // Handle to card  
    signed char hue // Hue level  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

hue

[in] Hue level

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Constants:

This function sets hue adjustment level.

Constant	Value	Description
VCEANCB_HueMax	127	Maximum hue level (equal to +178.6°)
VCEANCB_HueMin	-128	Minimum hue level (equal to -180.0°)
VCEANCB_HueDef	0	Medium hue level (equal to 0°)

See also:

[VCEANCB_Prepare](#), [VCEANCB_CameraConfig](#)

VCEANCB_setInputChannel

The **VCEANCB_setInputChannel** chooses input channel.

```
VCEANCB_Error VCEANCB_setInputChannel (  
    HANDLE hVCEANCB, // Handle to card  
    VCEANCB_SignalType signalType, // Type of signal  
    BYTE inputChannel // Number of input channel  
);
```

parameter:

hVCEANCB

[in] Handle to FrameGrabber.

signalType

[in] Type of signal (Composite or S-Video).

inputChannel

[in] Number of input channel

return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

marks:

This function selects input channel.

There are two types of signal:

VCEANCB_ST_Composite	Composite signal
VCEANCB_ST_SVideo	S-Video signal

There are two possible channels.

see also:

VCEANCB Prepare, VCEANCB CameraConfig,
VCEANCB SignalType

VCEANCB_setSaturation

The **VCEANCB_setSaturation** sets saturation adjustment level.

```
VCEANCB_Error VCEANCB_setSaturation (  
    HANDLE hVCEANCB, // Handle to card  
    signed char saturation // Saturation level  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

contrast

[in] Saturation level

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Remarks:

This function sets saturation adjustment level.

Constant	Value	Description
VCEANCB_SaturationMax	127	Maximum saturation level (equal to 1.984)
VCEANCB_SaturationMin	0	Minimum saturation level (equal to 0.0)
VCEANCB_SaturationDef	64	ITU saturation level (equal to 1.0)

See also:

[VCEANCB Prepare](#), [VCEANCB CameraConfig](#)

VCEANCB_SnapFrame

The **VCEANCB_SnapFrame** function snaps one frame from camera.

```
VCEANCB_Error VCEANCB_SnapFrame (  
    HANDLE hVCEANCB // Handle to card  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle is bad (possibly NULL).
VCEANCB_Err_notInitialized	FrameLink has not been successfully initialized.
VCEANCB_Err_notPrepared	VCEANCB Prepare was not successfully called.
VCEANCB_Err_bufferBusy	Snapping buffers are busy.
VCEANCB_Err_CamerasNotConnected	Camera is not connected.
VCEANCB_Err_lostContact	Contact with camera was lost during snapping.
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Remarks:

This function only snaps one frame to internal buffer, to get image call [VCEANCB_GetDIB](#) function.

Note: You have to call [VCEANCB_Prepare\(\)](#) function, before start snap.

See also:

[VCEANCB_Prepare](#), [VCEANCB_GetDIB](#), [VCEANCB_LockBuffer](#)

VCEANCB_SystemLastError

The **VCEANCB_SystemLastError** function returns value of **GetLastError()** occurred during last function returned **VCEANCB_Err_UnknownError**.

DWORD VCEANCB_SystemLastError ();

turn values:

Value of last error code

e also:

GetLastError()

VCEANCB_UnlockBuffer

The **VCEANCB_UnlockBuffer** function release buffer locked by **VCEANCB_LockBuffer()** function.

```
VCEANCB_Error VCEANCB_UnlockBuffer (  
    HANDLE hVCEANCB, // Handle to card  
    BYTE* pBuffer // Pointer to buffer  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

pBuffer

[out] Pointer to buffer.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle is bad (possibly NULL) or pBuffer was not produced by VCEANCB_LockBuffer function or was modified.
VCEANCB_Err_notGrabbed	VCEANCB_SnapFrame was not called.
VCEANCB_Err_bufferBusy	Snapping buffers are busy.

See also:

[VCEANCB_LockBuffer](#)

VCEANCB_GetTriggerStatus

The **VCEANCB_GetTriggerStatus** function query trigger for status.

```
VCEANCB_Error VCEANCB_GetTriggerStatus (  
    HANDLE hVCEANCB, // Handle to card  
    DWORD* pdwCurStatus, // Current status  
    DWORD* pdwWasRising, // Was state changed to rising  
    DWORD* pdwWasFalling, // Was state changed to falling  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

pdwCurStatus

[out] Current state of trigger

0 - Trigger released (No signal, Logic 0)

1 - Trigger pushed (Signal, Logic 1)

pdwWasRising

[out] Was state of trigger changed from Logic 0 to Logic 1 (pushed) since last function call

pdwWasFalling

[out] Was state of trigger changed from Logic 1 to Logic 0 (released) since last function call

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or any of pointers are bad.
---	---

See also:

VCEANCB_ConvertPixles2DIB

The **VCEANCB_ConvertPixles2DIB** function generates Device Independent Bitmap (DIB) from raw bytes recieved from FrameGrabber.

```
VCEANCB_Error VCEANCB_ConvertPixles2DIB (  
    HANDLE hVCEANCB, // Handle to card  
    BYTE* lpBuffer, // Handle to card  
    DWORD dwBufferSize, // Handle to card  
    HGLOBAL* phDIB // Pointer to DIB  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

lpBuffer

[in] Pointer to Raw bytes.

dwBufferSize

[in] Size of data.

phDIB

[out] Pointer to DIB handle.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or phDIB pointer is bad (possibly NULL).
VCEANCB_Err_notSupported	Unsupported camera operation mode.
VCEANCB_Err_noMemory	Not enough memory.

Remarks:

VCEANCB_GetDIB function generates Device Independent Bitmap explained in **Device-Independent Bitmaps** section of *Windows® GDI*. To free memory used by DIB call [VCEANCB_ReleaseDIB](#).

See also:

[VCEANCB_SnapFrame](#), [VCEANCB_ReleaseDIB](#),
[VCEANCB_DrawDIB](#), [VCEANCB_SaveDIBToFile](#),
[VCEANCB_LockBuffer](#)

VCEANCB_GrabFrame_Callback

The **VCEANCB_GrabFrame_Callback** user-defined callback function receives pixels buffer from grabbing stream.

```
typedef void (WINAPI *VCEANCB_GrabFrame_Callback)  
    (LPVOID lpPixelsBuffer, DWORD dwBufferSize, LPVOID lpUserData
```

User-defined function should be declared like this:

```
void WINAPI GrabFrame_Callback (  
    LPVOID lpPixelsBuffer // pointer to buffer with raw data  
    DWORD dwBufferSize // size of buffer  
    LPVOID lpUserData // User-defined data  
);
```

parameter:

lpPixelsBuffer

[in] Pointer to raw data received from digitizer.

dwBufferSize

[in] Size of buffer.

lpUserData

[in] User-defined data, specified in **VCEANCB_StartGrab** function.

marks:

Data format of pixel buffer explained in [VCEANCB_LockBuffer](#) function.

Note: Please notice, this function calls in different thread, then your main application.

see also:

[VCEANCB_PrepareGrab](#), [VCEANCB_StartGrab](#),
[VCEANCB_StopGrab](#), [VCEANCB_ConvertPixels2DIB](#),
[VCEANCB_LockBuffer](#)

VCEANCB_StartGrab

The **VCEANCB_StartGrab** function starts grabbing stream from camera.

```
VCEANCB_Error VCEANCB_StartGrab (  
    HANDLE hVCEANCB // Handle to card  
    VCEANCB_GrabFrame_Callback lpfnCallback // User-defined Callback f  
    LPVOID lpUserData // User-defined data  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

lpfnCallback

[in] User-defined call-back function to receive data from stream.

lpUserData

[in] User-defined data, which will be sent to *lpfnCallback* function.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle is bad (possibly NULL).
VCEANCB_Err_notInitialized	VCE-PRO has not been successfully initialized.
VCEANCB_Err_notPrepared	VCEANCB_PrepareGrab was not successfully called.
VCEANCB_Err_bufferBusy	Grabbing buffers are busy.
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Remarks:

This function starts grabbing stream and calls User-defined callback function, when new frame ready. **Note:** You have to call [VCEANCB_PrepareGrab\(\)](#) before start grabbing.

See also:

[VCEANCB_PrepGrab, VCEANCB_StopGrab,
VCEANCB_GrabFrame_Callback](#)

VCEANCB_StopGrab

The **VCEANCB_StopGrab** function stops grabbing stream.

```
VCEANCB_Error VCEANCB_StopGrab (  
    HANDLE hVCEANCB // Handle to card  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameLink card.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle is bad (possibly NULL).
VCEANCB_Err_notInitialized	FrameLink has not been successfully initialized.
VCEANCB_Err_notGrabbed	VCEANCB_StartGrab was not successfully called.
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

See also:

[VCEANCB_StartGrab](#)

VCEANCB_CopyDIB

The **VCEANCB_CopyDIB** function creates duplicate of Device Independent Bitmap (DIB).

```
VCEANCB_Error VCEANCB_GetDIB (  
    HGLOBAL hSrcDIB, // Source DIB  
    HGLOBAL* phDestDIB // Destination DIB  
);
```

Parameter:

hSrcDIB

[in] Handle to source DIB.

phDestDIB

[out] Pointer to destination DIB handle.

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hSrcDIB handle or phDestDIB pointer is bad (possibly NULL).
VCEANCB_Err_noMemory	Not enough memory.

Remarks:

To free memory used by DIB call [VCEANCB_ReleaseDIB](#).

See also:

[VCEANCB_GetDIB](#), [VCEANCB_ReleaseDIB](#), [VCEANCB_DrawDIB](#), [VCEANCB_SaveDIBToFile](#)

VCEANCB_SetColorKiller

The **VCEANCB_SetColorKiller** sets level of chrominance subcarrier to detect color.

```
VCEANCB_Error VCEANCB_SetColorKiller (  
    HANDLE hVCEANCB, // Handle to card  
    DWORD dwColorKiller // Killer level  
);
```

Parameter:

hVCEANCB

[in] Handle to FrameGrabber.

dwColorKiller

[in] Color Killer level

Return values:

If function succeeds, the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle or pCameraConfig pointer is bad (possibly NULL).
VCEANCB_Err_UnknownError	Error is unknown (possibly system error). To get extended error information, call VCEANCB_SystemLastError .

Remarks:

Set level of chrominance subcarrier to detect color.

0 - minimum level: color is switched on at low subcarrier levels

7 - recommended value

15 - maximum

See also:

VCEANCB_IsVideoSource

The **VCEANCB_IsVideoSource** function checks, is video source connected.

```
VCEANCB_Error VCEANCB_IsVideoSource (  
    HANDLE hVCEANCB // Handle to card  
);
```

Parameter:

hVCEANCB
[in] Handle to FrameLink card.

Return values:

If function succeeds, and video source is active the return value is [VCEANCB_Err_Success](#).

If function fails, the return value can be one of the following values.

VCEANCB_Err_badArgument	hVCEANCB handle is bad.
VCEANCB_Err_noVideoSource	Video source is inactive.

CEANCB Constants

The following constants are used with VCEANCB library:

[VCEANCB_Error](#)

[VCEANCB_VideoStandard](#)

[VCEANCB_VideoFreq](#)

[VCEANCB_SignalType](#)

VCEANCB_Error

[This is preliminary documentation and subject to change.]

```
typedef enum tagVCEANCB_Error {
    VCEANCB_Err_Success = 0, /* Success */
    VCEANCB_Err_noDevicePresent, /* No VCE ANCB Device present */
    VCEANCB_Err_DeviceBusy, /* VCE ANCB Device is busy */
    VCEANCB_Err_notInitialized, /* VCE ANCB SDK has not been ini
    VCEANCB_Err_noMemory, /* Not enough memory */
    VCEANCB_Err_badArgument, /* Bad argument has been passed */
    VCEANCB_Err_notPrepared, /* VCEANCB_Prepare was not successf
    VCEANCB_Err_notGrabbed, /* VCEANCB_SnapFrame was not succes
    VCEANCB_Err_bufferBusy, /* Buffers is busy */
    VCEANCB_Err_DeviceTimeout, /* device timeout error */
    VCEANCB_Err_noVideoSource, /* video source is not connected
    VCEANCB_Err_notSupported = 254, /* Operation or argument is
    VCEANCB_Err_UnknownError = 255, /* Error is not explained */
} VCEANCB_Error;
```

nstatnts

VCEANCB_Err_Success
No error occurred.

VCEANCB_Err_noDevicePresent
No VCE-PRO device present.

VCEANCB_Err_DeviceBusy
VCE-PRO device is already in-use.

VCEANCB_Err_notInitialized
FrameLink has not been successfully initialized.

VCEANCB_Err_noMemory
Not enough memory to perform operation.

VCEANCB_Err_badArgument
Bad (possibly NULL or pointed to NULL) argument passed.

VCEANCB_Err_notPrepared
[VCEANCB Prepare](#) was not successfully called.

VCEANCB_Err_notGrabbed
[VCEANCB SnapFrame](#) was not successfully called.

VCEANCB_Err_bufferBusy

Snapping buffers are busy. Check that you have unlock buffers.

VCEANCB_Err_DeviceTimeout

Device timeout error occurred. This error could happens if data, that sent to SnapData is invalid, or video source is not detected. It's recommended to re-prepare card and make sure, that video source is connected and active.

VCEANCB_Err_noVideoSource

Video source is not connected or not active. Check that power is turned on and channel selection is correct.

VCEANCB_Err_notSupported

Operation or argument is no supported.

VCEANCB_Err_UnknownError

Error is unknown (possibly system error). To get extended error information, call [VCEANCB_SystemLastError](#).

VCEANCB_SignalType

[This is preliminary documentation and subject to change.]

```
typedef enum tagVCEANCB_SignalType {
    VCEANCB_ST_Composite    = 0, // Composite signal
    VCEANCB_ST_SVideo      = 1, // S-Video signal
    VCEANCB_ST_unused      = 0xff, // Unused
} VCEANCB_SignalType;
```

VCEANCB_VideoStandard

[This is preliminary documentation and subject to change.]

```
typedef enum tagVCEANCB_VideoStandard {
    VCEANCB_VS_noSignal = 0, // No Signal or no color.
    VCEANCB_VS_NTSC = 1, // NTSC Signal
    VCEANCB_VS_PAL = 2, // PAL Signal
    VCEANCB_VS_SECAM = 3, // SECAM Signal
    VCEANCB_VS_unused = 0xff, // Unused
} VCEANCB_VideoStandard;
```

VCEANCB_VideoFreq

[This is preliminary documentation and subject to change.]

```
typedef enum tagVCEANCB_VideoFreq {  
    VCEANCB_VF_50 = 0, // 50Hz signal (Typicaly PAL or SECAM)  
    VCEANCB_VF_60 = 1, // 60Hz Signal (Typicaly NTSC)  
    VCEANCB_VF_unused = 0xffffffff, // Unused  
} VCEANCB_VideoFreq;
```

CEANCB Structures Reference

The following structures are used with VCEANCB library:

[VCEANCB_CameraConfig](#)

[VCEANCB_SnapData](#)

VCEANCB_CameraConfig

[This is preliminary documentation and subject to change.]

The VCEANCB_CameraConfig defines parameters that specifies camera

```
typedef struct tagVCEANCB_CameraConfig {
    DWORD cbSize; /* Size of structure */
    VCEANCB_VideoStandard videoStandard; // Video standard (NTSC;PAL;
    VCEANCB_VideoFreq videoFrequency; // Video Frequency (50Hz; 60Hz)

    VCEANCB_SignalType signalType; // Signal type (Composite;SVideo)
    DWORD inputChannel; // Channel number
    DWORD brightness;
    LONG contrast;
    LONG saturation;
    LONG hue;
    float gamma;
    VCEANCB_SnapData snapDataNTSC;
    VCEANCB_SnapData snapDataPAL;
} VCEANCB_CameraConfig;
```

Members

cbSize

Size of structure

videoStandard

Current videoStandard (read-only) (see also [VCEANCB_getVideoStandard](#))

videoFrequency

Current frequency of VideoSignal (readonly) (see also [VCEANCB_getVideoStandard](#))

signalType

Selected type of signal (see also [VCEANCB_setInputChannel](#))

inputChannel

Selected input channel (see also [VCEANCB_setInputChannel](#))

brightness

Selected brightness level (see also [VCEANCB_setBrightness](#))

contrast

Selected contrast level (see also [VCEANCB_setContrast](#))

saturation

Selected saturation level (see also [VCEANCB_setSaturation](#))

hue

Selected hue level (see also [VCEANCB_setHue](#))

gamma

Selected gamma level (see also [VCEANCB_setHue](#))

snapDataNTSC

Frame size for NTSC video (see also [VCEANCB_SnapData](#))

snapDataPAL

Frame size for PAL video (see also [VCEANCB_SnapData](#))

e Also:

[VCEANCB_Prepere](#)

VCEANCB_SnapData

[This is preliminary documentation and subject to change.]

The VCEANCB_SnapData defines parameters that specifies frame

```
typedef struct tagVCEANCB_SnapData
{
    DWORD cbSize;          /* Size of structure */
    DWORD xOffset;        /* X offset of active video area */
    DWORD xActive;        /* Width of active video area */
    DWORD xPixels;        /* Requested width in pixels of active video area */
    DWORD yOffset;        /* Y offset of active video area */
    DWORD yActive;        /* Height of active video area */
    DWORD yPixels;        /* Requested height in pixels of active video area */
    DWORD interlaced;     /* Flag to turn on interlaced snap */
    DWORD monochrome;     /* Flag to turn on monochrome snap */
} VCEANCB_SnapData;
```

e Also:

[VCEANCB Prepare](#), [VCEANCB CameraConfig](#)