ARTRAY Camera / Capture Module Software Developer Kit for TNIR

Dynamic Link Library for Windows XP,Vista,7,8,10
Functions Manual Version 1.3.0.0-4

Artray Co., Ltd.

Contents of DLL function

DLL Initializing

ArtCam_GetDllVersion

```
DWORD ArtCam_GetDllVersion(void)
```

Function: Obtain library's version

Argument: None

Function Detail:

Obtain version and type of DLL
Among returned DWORD (32 bits), DLL type is stored in upper 16 bits while DLL version is stored in lower 16 bits.
Before you use library, check the DLL versions you installed. So as SDK .

The version is obtained as 4 places integral number.
If the version is 1.278, 1278 is stored for lower 16 bits.

DLL types are as below:

CODE	DEVICE TYPE
ARTCAM_CAMERATYPE_008TNIR	ARTCAM-008TNIR
ARTCAM_CAMERATYPE_031TNIR	ARTCAM-031TNIR
ARTCAM_CAMERATYPE_0016TNIR	ARTCAM-0016TNIR
ARTCAM_CAMERATYPE_032TNIR_USB3_T2	ARTCAM-032TNIR
ARTCAM_CAMERATYPE_009TNIR_USB3_T2	ARTCAM-009TNIR
ARTCAM_CAMERATYPE_131TNIR_USB3_T2	ARTCAM-131TNIR
ARTCAM_CAMERATYPE_990SWIR	ARTCAM-990SWIR
ARTCAM_CAMERATYPE_990SWIR_TEC	ARTCAM-990SWIR-TEC
ARTCAM_CAMERATYPE_991SWIR	ARTCAM-991SWIR
ARTCAM_CAMERATYPE_991SWIR_TEC	ARTCAM-991SWIR-TEC

ArtCam_GetLastError

```
LONG  ArtCam_GetLastError(HACAM hACam)
```

Function: Obtained error

Argument:

HACAM	*hACam*	Handle for distinguish cameras

Function Detail:

When error occurs in return value of function, please call this function to obtain details of error.
Error is stored in stack type of data configuration.
Errors can be called in sequential order.

ERROR CODE	ERROR DETAIL
ARTCAMSDK_NOERROR	Normal
ARTCAMSDK_NOT_INITIALIZE	Not Initialized
ARTCAMSDK_DISABLEDDEVICE	Tray to access to unusable device
ARTCAMSDK_CREATETHREAD	Failureure to create a thread for image capture
ARTCAMSDK_CREATEWINDOW	Failureure to create a window
ARTCAMSDK_OUTOFMEMORY	Not enough memory for image transferring. Or Failureure to obtain memory
ARTCAMSDK_CAMERASET	Error at camera (device) setting
ARTCAMSDK_CAMERASIZE	Error at camera (device) size setting
ARTCAMSDK_CAPTURE	Failureure at image capture
ARTCAMSDK_PARAM	Wrong argument
ARTCAMSDK_DIRECTSHOW	DirectShow Initializing error
ARTCAMSDK_UNSUPPORTED	This function is not supported
ARTCAMSDK_UNKNOWN	Unidentified error
ARTCAMSDK_CAPTURELOST	Lost device
ARTCAMSDK_FILENOTFOUND	Cannot find specified file
ARTCAMSDK_FPGASET	Error at FPGA setting
ARTCAMSDK_TRANSIMAGEFAILED	Failure of image transferring

ArtCam_Initialize

```
HACAM ArtCam_Initialize(HWND hWnd)
```

Function: Initialize library

Argument:

HACAM	*hACam*	Handle for distinguish cameras

Function Detail:

Initialize library.
Call this function first when you use this library

Once this function is succeeded, handle for camera identification is obtained in return value.
On the other hand, if it is Failureed, NULL or 0 is returned.

By setting window handle to hWnd, WM_ERROR is sent to window procedure when an error occurs.

Also whenever this function is called, the last parameter setting is read from registry.
Each parameter setting is saved under the below registry key.
(Some parameters are not saved)
HKEY_CURRENT_USER¥Software¥Artray¥ArtCam[MODEL NAME]Sdk

ArtCam_Release

```
BOOL ArtCam_Release(HACAM hACam)
```

Function: Release library

Argument:

HACAM	*hACam*	Handle for distinguish cameras

Function Detail:

Release all plugged cameras, and initialize all data within class.
Call this function when you end application or stop operation of cameras.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

To display image again, call ArtCam_Initialize.

Also whenever this function is called, the last parameter setting is read from registry.
(Some parameters are not saved)

Image capture

ArtCam_Preview

```
BOOL  ArtCam_Preview(HACAM hACam)
```

Function: Display image

Argument:

HACAM	*hACam*	Handle for distinguish cameras

Function Detail:

Image display is controlled by SDK.
Call ArtCam_Initialize before using this function.
When this function succeeds, create a sub-window within the window specified by ArtCam_SetPreviewWindow. Image will be displayed in the sub-window.
If setting is not done by ArtCam_SetPreviewWindow, new window will be created, and image will be displayed.
Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_Record

```
BOOL ArtCam_Record(
    HACAM hACam, LPCTSTR lpAviName, UINT RecTime,BOOL fShow)
```

Function: Record to file

Argument:

HACAM	*hACam*	Handle for distinguish cameras
LPCTSTR	*lpAviName*	Name of file to be saved
UINT	*RecTime*	Recording time (milli-second) Continuous recording at 0
BOOL	*fShow*	Display image or not

Function Detail:

When RecTime is specified, recording will automatically end as time out. However, device will not be released, and therefore image will still be displayed.
If you like to execute some process at the end of recordings, you need to obtain timing using timer. Regarding fShow, hiding image will prevent loss of frames.
Recorded files will be saved in uncompressed AVI format. When recorded file size exceeds 4GB, there might be problem playing the file in program like Windows Media Player.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0
Remark: This function is exclusively for ArtCamSdk.dll (Direct Show Camera)

ArtCam_CallBackPreview

```
BOOL ArtCam_CallBackPreview(
    HACAM hACam,
    HWND hWnd, LPBYTE lpImage, LONG Size, BOOL TopDown)
```

Function:Obtain image data while display live video

Argument:

HACAM	*hACam*	Handle for camera identification
HWND	*hWnd*	Window Handle for receiving message
LPBYTE	*lpImage*	Address of arrangement for receivingnimage data
LONG	*Size*	Arrangement length of lpImage
BOOL	*TopDown*	Determine whether image is up or down

Function Detail:

When hWnd is specified to window handle, WM_GRAPHPAINT is sent to specified window procedure.

When lpImage and Size are specified, image is copied to the alignment, which was specified at lpImage before WM_GRAPHPAINT.
Image will not be copied unless the size of alignment is equal to or larger than size of image.
Do not insert address of temporary alignment to lpImage.

If bitmap prepared is DDB (top-down), specify Topdown as True.
If bitmap is DIB (bottom-up), specify Topdown as False.

Like ArtCam_Preview, this function also has automatic display. Procedure for auto-display is same as that of ArtCam_Preview.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

*1: It is relatively difficult to obtain message with VB. There may be error due to processing speed of VB. Although the function itself can be used, real-time processing by WM_GRAPHPAINT should be avoided.
(With the current sample, that procedure is removed, and timing of display is controlled by timer)

ArtCam_SnapShot

```
BOOL ArtCam_SnapShot(
    HACAM hACam, LPBYTE lpImage, LONG Size, BOOL TopDown)
```

Function: Obtain image of camera only once

Argument:

HACAM	*hACam*	Handle for camera identification
LPBYTE	*lpImage*	Address of arrangement for receiving image data
LONG	*Size*	Arrangement length of lpImage
BOOL	*TopDown*	Determination of ups and down of image

Function Detail:

Obtained only 1 image from a camera by soft trigger.
When function succeeds, obtained data is stored in lpImage.
Image will not be obtained unless the size of alignment is equal to or larger than size of image.

If bitmap prepared is DDB (top-down), specify Topdown as True. If bitmap is DIB (bottom-up), specify Topdown as False.

While ArtCam_GetImage captures a frame in preview mode, ArtCam_SnapShot captures a frame in non-preview mode.

This function will Failure if preview is displayed with other functions such as ArtCam_Preview

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_Capture

```
BOOL ArtCam_Capture(HACAM hACam)
```

Function:Initialize camera for continuous snapshot

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

Initialize camera to use ArtCam_SnapShot continuously.
Normally when ArtCam_SnapShot is used, procedure proceeds as following:
Initialize - Obtain - Release
However, if you initialize beforehand with this function, the process of "Initialize" and "Release" will be ignored when ArtCam_SnapShot is called. Hence the image can be obtained with high-speed.
To stop ArtCam_SnapShot and release camera, call ArtCam_Close.

The main flow is as following:

ArtCam_Capture (Initialize) -> ArtCam_SnapShot (Can be used unlimitedly) ->
ArtCam_Close (Release)

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_Close

```
BOOL  ArtCam_Close(HACAM hACam)
```

Function: Release device

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail :

Stop preview screen, and release deivce. Use this function to release device when you obtain images with the following functions.
ArtCam_Preview
ArtCam_Record
ArtCam_CallBackPreview
ArtCam_Capture
ArtCam_Trigger

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_Trigger

```
BOOL ArtCam_Trigger(
    HACAM hACam,
    HWND hWnd, LPBYTE lpImage, LONG Size, BOOL TopDown)
```

Function:Obtain image of camera in external trigger mode

Argument:

HACAM	*hACam*	Handle for camera identification
HWND	*hWnd*	Window Handle for receiving message
LPBYTE	*lpImage*	Address of arrangement for receiving image data
LONG	*Size*	Arrangement length of lpImage
BOOL	*TopDown*	Determination of up and down of image

Function Detail:

Procedure of this function is similar to that of ArtCam_CallBackPreview.
Timing of capturing depends on camera's closck speed with ArtCam_CallBackPreview.
With this function, capturing is processed when triggered with external trigger.

When you initialize with this function, updates and obtaining message of image is sent only after the trigger is sent to camera.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

*1: Use pulse signal from 0-5V to 0-12V for trigger signal.
Timing from input trigger to take a picture is different from each cameras.
Please refer the manual for more detail.

WM_GRAPHPAINT

```
#define  WM_GRAPHPAINT      WM_APP + 2
```

Function: Message is issued when a camera image is updated.

WPARAM	*wParam*	LPGP_INFO
LPARAM	*lParam*	Always NULL

Function Detail:

LPGP_INFO in which is received by *wParam* is pointer to the structure that stores image data

```
typedef struct GP_INFO {
    LONG        lSize;
    LONG        lWidth;
    LONG        lHeight;
    LONG        lBpp;
    LONG        lFps;
    LPBYTE      pImage;
} *LPGP_INFO;

LPGP_INFO lpGPIF = (LPGP_INFO)wParam
```

This message is sent to the callback procedure of the window when Window Handle is set to hWnd at ArtCam_CallBackPreview and ArtCam_Trigger.

This message is sent when image is updated.

To obtain image data, assign pointer and array length of alignment to lpImage and Size of ArtCam_CallBackPreview. Then image data is stored in specified alignment
when this message is sent.

When wParam is NULL, WM_GRAPHPAINT becomes error.
wParam and lParam will mean WM_ERROR.

WM_GRAPHPAINT is defined as 0x8002

WM_ERROR

```
#define  WM_GRAPHPAINT      WM_APP + 3
```

Function: Receive error message

WPARAM	*wParam*	Always 0
LPARAM	*lParam*	Error Code

Function Detail:

When Window Handle is specified at ArtCam_Initialize, error code is sent to Window Procedure in case error occurs within SDK.

WM_ERROR is defined as 0x8003.

Error codes are as below:

ERROR CODE	STATUS
ARTCAMSDK_NOERROR	Normal
ARTCAMSDK_NOT_INITIALIZE	not initialized
ARTCAMSDK_DISABLEDDEVICE	It was going to access disable device
ARTCAMSDK_CREATETHREAD	Failureure of creating thread for capturing
ARTCAMSDK_CREATEWINDOW	Failureure of creating window
ARTCAMSDK_OUTOFMEMORY	No enough memory for transferring image
	Or Failureure of securing memory
ARTCAMSDK_CAMERASET	Error of camera (device) settings
ARTCAMSDK_CAPTURE	Failureure of cap
ARTCAMSDK_PARAM	Wrong argument
ARTCAMSDK_DIRECTSHOW	Error of DirectShow initialization
ARTCAMSDK_UNSUPPORTED	This function is not supported
ARTCAMSDK_UNKNOWN	Unknown error
ARTCAMSDK_CAPTURELOST	Device lost
ARTCAMSDK_FILENOTFOUND	Cannot find specified file
ARTCAMSDK_FPGASET	Error at FPGA setting

ArtCam_StartPreview

```
BOOL ArtCam_StartPreview(HACAM hACam)
```

Function: Start preview

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

Start preview of camera image
This function is used internally for ArtCam_Preview, ArtCam_Record and ArtCam_CallBackPreview.
This function is only used to regenerate image, in which preview is stop, by calling ArtCam_StopPreview.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_StopPreview

```
BOOL ArtCam_StopPreview(HACAM hACam)
```

Function: Stop preview

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

This function stops preview of image.
This function does not release device.
Please use this function only when you need to stop preview temporarily.

To display preview again, use ArtCam_StartPreview.

This function is only available when preview is displayed with ArtCam_Preview and ArtCam_CallBackPreview.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_SaveImage

```
BOOL ArtCam_SaveImage(
    HACAM hACam, LPCTSTR lpSaveName, LONG FileType)
```

Function:Save image of camera

Argument:

HACAM	*hACam*	Handle for camera identification
LPCTSTR	*lpSaveName*	Name of file to be saved
LONG	*FileType*	Type of save

Function Detail:

Save camera image in computer files
Image to be saved is the last image obtained by image-capturing functions such as ArtCam_Preview, ArtCam_CallBackPreview, ArtCam_SnapShot & ArtCam_Trigger

Please not that depending on system environment, speed clock of camera and file types, saved image may deteriorate while real-time image is obtained with ArtCam_Preview and ArtCam_CallBackPreview.
When this happens, stopping image update temporarily by ArtCam_StopPreview may prevent image deterioration.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

This function is used to save live image obtained by camera.
To save images that are processed by application, save the images at application.
File type can be selected from BMP, binary (RAW), JPEG (high-quality, standard & low-quality), PNG and TIFF.

You cannot save 16 bits image in JPEG
When you save image in JPEG, it is saved as gray scale of 8 bits bit-depth.
BMP and RAW can be saved in 16 bits. However, pallet info is not saved in file. Therefore image may not be display correctly for softwares that do not have special reading routine.

To save images in 16 bits, use of PNG and TIFF are recommended. With these file Initializes, we recommend you to read the images in Artray's Viewer Software or Adobe Photoshop6.
Please note that not every image-processing application is compatible with 16 bits image.

Regarding files to be saved with this function, we only support on reading procedure on BMP and RAW.

We will not provide suppont on reading procedures of other file Initializes and saving procedure.

ArtCam_GetImage

```
BOOL ArtCam_GetImage(
    HACAM hACam, LPBYTE lpImage, LONG Size, BOOL TopDown)
```

Function: Obtain image of camera

Argument:

HACAM	*hACam*	Handle for camera identification
LPBYTE	*lpImage*	Address of arrangement for receiving image data
LONG	*Size*	Arrangement length of lpImage
BOOL	*TopDown*	Determination of up and down of image

Function Detail:

Obtain image of camera.
When function succeeds, previously obtained data is stored in lpImage.
Image will not be obtained unless the size of alignment is equal to or larger than size of image.
If bitmap prepared is DDB (top-down), specify Topdown as True. If bitmap is DIB (bottom-up), specify Topdown as False.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

This function is used to obtain image asynchronously while ArtCam_Preview or ArtCam_CallBackPreview is used.
If you only need to obtain 1 frame, use ArtCam_SnapShot.
This function assumes that the PC with low specs is used, or language, which has slow processing speed, is used.
To create with C & C++, receive message of image updates by WM_GRAPHPAINT

Setting dialog

ArtCam_SetCameraDlg

```
BOOL ArtCam_SetCameraDlg(HACAM hACam, HWND hWnd)
```

Function: Show dialog of camera settings

Argument:

HACAM	*hACam*	Handle for camera identification
HWND	*hWnd*	Parent window for showing dialog

Function Detail:

This function displays a dialog box that allows you to alter settings such as size of image and frame rate.
Dialog box displayed varies with the device plugged.
When you call this function while preview is displayed, preview will temporarily stop.
Preview will be displayed again once dialog box is closed.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_SetImageDlg

```
BOOL ArtCam_SetImageDlg(HACAM hACam, HWND hWnd)
```

Function: Show dialog of filter settings

Argument:

HACAM	*hACam*	Handle for camera identification
HWND	*hWnd*	Parent window for showing dialog

Function Detail:

This function displays a dialog box that allows you to alter settings such as brightness, contrast and white balance.
Dialog box displayed varies with the device plugged.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_SetAnalogDlg

```
BOOL ArtCam_SetCameraDlg(HACAM hACam, HWND hWnd)
```

Function: Show dialog of port/camera settings

Argument:

HACAM	*hACam*	Handle for camera identification
HWND	*hWnd*	Parent window for showing dialog

Function Detail:

This function displays a dialog box that allows you to alter settings such as analog port and internal camera device.
Dialog box displayed varies with the device plugged.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_SetPreviewWindow

```
BOOL ArtCam_SetPreviewWindow(
    HACAM hACam,
    HWND hWnd, LONG Left, LONG Top, LONG Right, LONG Bottom)
```

Function: Specify window to display image of camera and specify its range

Argument:

HACAM	*hACam*	Handle for camera identification
HWND	*hWnd*	Specify handle of window to be displayed
LONG	*Left*	Specify upper-left X-coordinate of rectangle
LONG	*Top*	Specify upper-left Y-coordinate of rectangle
LONG	*Right*	Specify lower-right X-coordinate of rectangle
LONG	*Bottom*	Specify lower-right Y-coordinate of rectangle

Function Detail:

When Window handle is specified to hWnd, create child window in the window and display in the child window.

When NULL is specified to hWnd, create new window.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_SetCaptureWindowEx

```
BOOL ArtCam_SetCaptureWindowEx(
    HACAM hACam,
    LONG HTotal, LONG HStart, LONG HEffective,
    LONG VTotal, LONG VStart, LONG VEffective)
```

Function: Specify image size of camera(ROIFunction)

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*HTotal*	Specify total horizontal width of camera in the unit of pixel
LONG	*HStart*	Specify starting point of horizon
LONG	*HEffective*	Specify effective horizontal width in the unit of pixel
LONG	*VTotal*	Specify vertical total height in the unit of pixel
LONG	*VStart*	Specify starting point of vertical
LONG	*VEffective*	Specify effective vertical height

Function Detail:

Set up capture image size.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ROIFunction is a function only for CMOS sensor camera.
You cannot set up image size at CCD sensor camera.

For color image, because of Bayer converting, you need more than 5 pixel active imager size at both horizontal and vertical.

We recommend to set up multiple of 4 for active horizontal pixel and active vertical pixel. Especially, it would not view images properly if you set up other than multiple of 4 to active horizontal pixel.

The setting value of this function will not be saved in the registry automatically.
To save the size settings automatically, please call the function “ArtCam_SetCameraDlg” and set the size in the dialog window.

ARTCAM-131TNIR is compatible with 3 size. Please choose the parameter from the table below:

Image Size	***HTotal***	***HStart***	***HEffective***	***VTotal***	***VStart***	***VEffective***
640 * 512	640	0	640	512	0	512
320 * 240	640	160	320	512	136	240
160 * 120	640	240	160	512	196	120

ArtCam_GetCaptureWindowEx

```
BOOL ArtCam_GetCaptureWindowEx(
    HACAM hACam,
    LONG* HTotal, LONG* HStart, LONG* HEffective,
    LONG* VTotal, LONG* VStart, LONG* VEffective)
```

Function: Obtain image size of camera

Argument:

HACAM	*hACam*	Handle for camera identification
LONG*	*HTotal*	Returns total width of camera in unit of pixel
LONG*	*HStart*	Returns starting point of width
LONG*	*Heffective*	Returns operative width of camera
LONG*	*VTotal*	Returns total height of camera in unit of pixel
LONG*	*VStart*	Returns starting point of height
LONG*	*VEffective*	Returns operative height of camera

Function Detail:

For ArtCamSdk.dll :

Use ArtCam_Width ArtCam_Height ArtCam_Fps

Other :

Obtain each parameter of camera

Success: Returned TRUE or 1

Failure: Returned FALSE or 0

ArtCam_SetColorMode

BOOL ArtCam_SetColorMode(HACAM hACam, LONG ColorMode)

Function: Set color mode for image capturimg

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*ColorMode*	Number of data bits

Function Detail:

Specify number of bits.

8: 8 bits monochrome image
16 16 bits monochrome image
24: 24 bits color image (BGR, 8 bits each)
32: 32 bits color image (BGRA, 8 bits, A=invalid)
48: 48 bits color image (BGR, 16 bits each)
64: 64 bits color image (BGRA, 16 bits each, A=invalid)

With 16 (10) bits, numerical values vary slightly depending on the environment.
When you create an application, make sure that the application is compatible with 10, 12, 14, & 16.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_GetColorMode

```
LONG ArtCam_GetColorMode (HACAM hACam)
```

Function: Obtain current color mode

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

Success: Returned image bit number as LONG value (8 - 64)

Failure: Returned -1

Under 8, 24, 32bit color mode, the return value will be accordingly 8, 24, 32bit.
Under 16bit color mode, the return value of 10bit camera will be 10bit; the return value of 12bit camera will be 12bit.
Under 48bit color mode, the return value of 10bit camera will be 42bit; the return value of 12bit camera will be 44bit.
Under 64bit color mode, the return value of 10bit camera will be 58bit; the return value of 12bit camera will be 60bit.

The relationship between color mode and return value

	16bit Mode	48bit Mode	64bit Mode
10bit Camera	10	42	58
12bit Camera	12	44	60
14bit Camera	14	46	62
16bit Camera	16	48	64

ArtCam_SetDeviceNumber

```
BOOL ArtCam_SetDeviceNumber(HACAM hACam, LONG Number)
```

Function: Assign number of device to be plugged

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*Number*	Assign device number from 0 to 9

Function Detail:

After you call this function and initialize with functions such as ArtCam_ Preview, ArtCam_ Record & ArtCam_CallBackPreview, image of specified device will be displayed. To confirm device number, use ArtCam_EnumDevice & ArtCam_GetDeviceName.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_GetDeviceName

```
BOOL ArtCam_GetDeviceName(
    HACAM hACam, LONG index, LPSTR szDeviceName, LONG nSize)
```

Function: Obtain name of specified device

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*index*	Specify number of device from 0 to 9
LPSTR	*szDeviceName*	Names of devices are copied, if they are operative
LONG	*nSize*	Size of szDeviceName

Function Detail:

Confirm if device specified by index is operative. If it's operative, store the name of device to szDeviceName.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_EnumDevice

```
LONG ArtCam_EnumDevice(
    HACAM hACam, TCHAR szDeviceName[10][256])
```

Function: Recount names of operative device

Argument:

HACAM	*hACam*	Handle for camera identification
TCHAR	*szDeviceName*	Names of operative devices are copied

Function Detail:

Utilized device name is stored in *szDeviceName.*
For example, if two ARTCAM-008TNIR cameras are possible to use,
Normaly,
a string ArtCam008TNIR _0 is stored in *szDeviceName*[0]
a string ArtCam008TNIR _1 is stored in *szDeviceName*[1]

Number specified by ArtCam_SetDeviceNumber is same as alignment number stored in *szDeviceName*.
To use device stored in ArtCam008TNIR _1, specify ArtCam_SetDeviceNumber (1).

If the function is successfully worked, utilized device number is returned as LONG value.

Please give the strings of [10][256] for second Argument.
If a string is smaller than this, Return value would be returned to 0.

You cannot use this function for VB.NET and C#.NET.
To obtain device name with other languages, please use ArtCam_GetDeviceName

ArtCam_Width

```
LONG ArtCam_Width(HACAM hACam)
```

Function: Obtain width of camera image

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

Cameras that capture size is fixed :
Standard size set within SDK is returned by LONG value.

Cameras that capture size is flexible :
Assigned Heffective value at ArtCam_SetCaptureWindowEx is returned by LONG value.

ArtCam_Height

```
LONG ArtCam_Height(HACAM hACam)
```

Function: Obtain height of camera image

Argument:

HACAM	*hACam*	Handle of camera identification

Function Detail:

Cameras that capture size is fixed :
Standard size set within SDK is returned by LONG value.

Cameras that capture size is flexible :
Assigned Veffective value at ArtCam_SetCaptureWindowEx is returned by LONG value.

ArtCam_GetCameraInfo

```
BOOL ArtCam_GetCameraInfo(HACAM hACam, LPCAMERAINFO pInfo)
```

Function: Obtain camera information

Argument:

HACAM	*hACam*	Handle for camera identification
LPCAMERAINFO	*pInfo*	Camera information

Function Detail:

Obtaing the information of connecting camera's setting possible value.

CAMERAINFO type structure is defined as below:

```
CAMERAINFO {
    LONG  lSize;          // structure's size
    LONG  lWidth;         // Camera's effective maximum width
    LONG  lHeight;        // Camera's effective maximum height
    LONG  lGlobalGainMin; // Lowest value of global gain
    LONG  lGlobalGainMax; // Maximum value of global gain
    LONG  lColorGainMin;  // Lowest value of color gain
    LONG  lColorGainMax;  // Maximum value of color gain
    LONG  lExposureMin;   // Lowest value of exposure time
    LONG  lExposureMax;   // Maximum value of exposure time
    double  dExposureMin; // Lowest value of exposure time[sec]
    double  dExposureMax; // Maximum value of exposure time[sec]
} *LPCAMERAINFO;
```

ArtCam_SetIOPort

```
BOOL ArtCam_SetIOPort(
    HACAM hACam, BYTE byteData, LONG longData, DWORD Reserve)
```

Function: Write data to IO.

Argument:

HACAM	*hACam*	Handle for camera identification
BYTE	*byteData*	Data written in IO (byte data)
LONG	*longData*	Not in use. Please specify 0.
DWORD	*Reserve*	Not in use. Please specify 0.

Function Detail:

Write data (8 bit) into I/O port.
Port will be initialized at low level when loading device driver (i.e. loading operating system or plugging USB)
e.g. When "0x0C" is saved into "byteData", both OUT0 and OUT1 ports will be at Hi level.
Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_GetIOPort

```
BOOL ArtCam_GetIOPort(
    HACAM hACam, LPBYTE byteData, LPLONG longData, DWORD Reserve)
```

Function: Read data from IO

Argument:

HACAM	*hACam*	Handle for camera identification
LPBYTE	*byteData*	Data read from IO (byte data)
LPLONG	*longData*	Not in use. Please specify NULL.
DWORD	*Reserve*	Not in use. Please spegify 0.

Function Detail:

Read data (8 bit) from I/O port. At the newest driver (July 11th. 2007) Low level (LSB)'s 1and 2 bits are IN0, and IN1 port.
Input level of port returns 1 when it is Hi in byteData.
e.g. When IN0 level is Low and IN1 level is Hi, "0x02" will be in byteData.
Success: Returned TRUE or 1
Failure: Returned FALSE or 0

This function is effective only for corresponded to I/O customized camera.
Port is Initialized to Low level at the time device drive is loading (Start OS or connecting USB). Voltage is unstable till driver is loaded.

ArtCam_SetSubSample

```
BOOL ArtCam_SetSubSample(HACAM hACam, LONG SubSampleMode)
```

Function: Set sub-sampling mode

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*SubSampleMode*	Sub-sampling mode

Function Detail:

This function sets sub-sampling transfer mode.
Thinning out image is transferred. Image is thinned out by value set in *SubSampleMode.*

SUBSAMPLE_1	All data
SUBSAMPLE_2	Data equals to half of matrix
SUBSAMPLE_4	Data equals to quarter of matrix
SUBSAMPLE_8	Data equals to eighth of matrix
BINNING_2	1/2 binning
BINNING_4	1/4 binning

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

When ArtCam_CallBackPreview is used in this function, data less than assigned image size is transferred.

Transfer mode is different at each models.
There is not this function at CCD camera.
The setting value of this function will not be saved in the registry automatically.
To save the size settings automatically, please call the function ArtCam_SetCameraDlg and set the size in the dialog window.

ArtCam_GetSubSample

```
LONG ArtCam_GetSubSample(HACAM hACam)
```

Function: Obtain current sub-sampling mode

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

Obtain current pixel skipping transfer mode.
Return -1 if the function is Failureed.

ArtCam_SetWaitTime

```
BOOL ArtCam_SetWaitTime(HACAM hACam, LONG WaitTime)
```

Function: Assign WaitTime

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*WaitTime*	WaitTime

Function Detail:

This function assigns waiting time for obtaining video from ArtCam_Preview and ArtCam_CallBackPreview.
Specified wait time between frame by mm/sec. Default is 10.

Success: Returned TRUE or 1.
Failure: Returned FALSE or 0.

Frame rate will be increased when you assign a small value in Wait Time. Missing will be decreased.
CPU's using rate will be increased.
Please specify between 5 to 20 as average number.
Frame rate will decrease when you assign a large value for Wait Time

The setting value of this function will not be saved in the registry automatically.
To save the size settings automatically, please call the function ArtCam_SetCameraDlg and set the size in the dialog window.

ArtCam_GetWaitTime

```
LONG ArtCam_GetWaitTime(HACAM hACam)
```

Function: Obtain WaitTime

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

Success: Obtain current Wait Time by LONG value to RETURN value.
Failure: Return -1 to RETURN value.

ArtCam_SetMirrorV

```
BOOL ArtCam_SetMirrorV(HACAM hACam, BOOL Flg)
```

Function: Set flip vertical mirroring function

Argument:

HACAM	*hACam*	Handle for camera identification
BOOL	*Flg*	Reverse flag

Function Detail:

With camera's hardware function, you can transfer data in flip vertical.
Setting *Flg* to True will enable mirroring function, while false will disable the function.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

Each models has different default flg.

ArtCam_GetMirrorV

```
BOOL ArtCam_GetMirrorV(HACAM hACam)
```

Function:Obtain conditions of flip vertical mirroring function

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

Confirm if flip vertical mirroring function is enabled or not.

Mirroring function enabled: True
Mirroring function disabled: False

Each models has different default flg.

ArtCam_SetMirrorH

```
BOOL ArtCam_SetMirrorH(HACAM hACam, BOOL Flg)
```

Function:Set flip horizontal mirroring function

Argument:

HACAM	*hACam*	Handle for camera identification
BOOL	*Flg*	Reverse flag

Function Detail:

With camera's hardware function, you can transfer data in flip horizontal. Setting Flg to True will enable mirroring function, while false will disable the function.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

Each models has different default flg.

ArtCam_GetMirrorH

```
BOOL ArtCam_GetMirrorH(HACAM hACam)
```

Function:Obtain current conditions of flip horizontal mirroring function

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

With camera's hardware function, you can transfer data in flip horizontal.

Mirroring function enabled: True
Mirroring function disabled: False

Each models has different default flg.

ArtCam_SetAutoIris

```
BOOL ArtCam_SetAutoIris(HACAM hACam, LONG Value)
```

Function: Set condition of auto-iris

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*Value*	Flag for auto-iris

Function Detail:

Set up Auto Iris (Auto brightness revision)'s effectrive/invalid.
specify in *Value*:

Value =0 Invalid *Auto Iris*
Value=1 Effective Auto Iris by Shutter Speed
Value=2 Effective Auto Iris by Global Gain

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

There are some difference depend on using environment by each model.
We recommend to avoid using this function with filter because if filter of Sharpness or Brightness is effective, this function would not work properly.

ArtCam_GetAutoIris

```
LONG ArtCam_GetAutoIris(HACAM hACam, LPBOOL Error)
```

Function: Obtain condition of auto-iris

Argument:

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Error information

Function Detail:

Obtain current condition of auto-iris (Auto brightness revision) by LONG value.
Success: Invalid=0
Exposure time settings=1
Gain settings=2

Succsess: TRUE on *Error*
Failure: FALSE on *Error*

ArtCam_WriteRegistor

```
BOOL ArtCam_WriteRegister(
    HACAM hACam, BYTE Address, DWORD Value)
```

Function: Write to a sensor register

Argument:

HACAM	*hACam*	Handle for camera identification
BYTE	*Address*	Writing address
DWORD	*Value*	Writing data

Function Detail:

Writing data on camera sensor's register

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

We do not open the detail of register setting on public.
This function is for some customized cameras.
Please avoid to change a register value on normal camera. Unexpected trouble would be occurred.

ArtCam_ReadRegister

```
DWORD ArtCam_ReadRegister(
    HACAM hACam, BYTE Address, LPBOOL Error)
```

Function: Read sensor register value

Argument:

HACAM	*hACam*	Handle for camera identification
BYTE	*Address*	Reading address
LPBOOL	*Error*	Error information

Function Detail:

Read register value or a camera sensor's specified address.

Succsess: TRUE on *Error*
Failure: FALSE on *Error*

We do not open the register setting detail on public.
This function is only for some customized cameras.

ArtCam_SetFilterValue

```
BOOL ArtCam_SetFilterValue(
    HACAM hACam, LONG FilterType, LONG Value)
```

Function:Set image filter information

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*FilterType*	Type of filter to be set
LONG	*Value*	Number to be set

Function Detail:

This function allows you to directly set values, which can be set with ArtCam_SetImageDlg.
Regarding FilterType, please refer to defined file of each language.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_GetFilterValue

```
LONG ArtCam_GetFilterValue(
    HACAM hACam, LONG FilterType, LPBOOL Error)
```

Function: Obtain image filter information

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*FilterType*	Type of filter to be set
LPBOOL	*Error*	Error information

Function Detail:

This function allows you to obtain value of parameter that can be set ArtCam_SetImageDlg and ArtCam_SetAnalogDlg.
When you set NULL for Error, error info will not be obtained.
Regarding FilterType, please refer to defined file of each language.

Succsess: TRUE on *Error* and returned setting number on *Filter Type* to LONG value.
Failure: FALSE on *Error*

ArtCam_Set***

```
BOOL ArtCam_Set***(HACAM hACam, LONG Value)
```

Function: Set image filter information

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*Value*	Value to be set

Function Detail:

Wrapper function of ArtCam_SetFilterValue
This function will be called when second argument of ArtCam_SetFilterValue is set

For example, if you want to change Global Gain to 30,

Instead of doing ArtCam_SetFilterValue
(hACam, ARTCAM_FILTERTYPE_GLOBAL_GAIN, 30),
set to ArtCam_SetGlobalGain(hACam,30).

ArtCam_Get***

```
LONG ArtCam_Get***(HACAM hACam, LPBOOL Error)
```

Function: Obtain image filter information

Argument:

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Error information

Function Detail:

Wrapper function of ArtCam_GetFilterValue
This function will be called when second argument of ArtCam_GetFilterValue is set

ArtCam_GetRealExposureTime

```
DWORD ArtCam_GetRealExposureTime(HACAM hACam, LPBOOL Error)
```

Function: Get the real exposure time

Argument:

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Error information

Function Detail:

Obtain the real exposure time by LONG value.
The unit of exposure time is 100 microsecond.

Success: TRUE on Error.
Failure: FALSE on Error.

Notice:

Please note that the unit of a related function ArtCam_SetExposureTime is in H because ArtCam_SetExposureTime sets the exposure time on the sensors. H is a unit of shutter speed calculation, not a time unit. To obtain the shutter speed in time units, please use ArtCam_GetExposureTime.

The exposure time is calculated by the following formulas.

1H = (Effective Horizontal Pixels + Horizontal Blank Pixels) * Pixel Clock
Exposure Time = Shutter Setting Value * 1H

This function calculates internally on the software.

Please note that the pixel clock varies with the model. For example, the clock parameter for 130MI is 1/24000000. Also, if the hardware has been updated or the clock is set to half, this function may not return the correct value.

ArtCam_SetRealExposureTime

```
BOOL ArtCam_SetRealExposureTime(HACAM hACam, LONG Value)
```

Function: Set the real exposure time

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*Value*	Value to be set

Function Detail:

This function sets the real exposure time by LONG value.
The unit of exposure time is 100 microsecond.
Success: Returned TRUE or 1
Failure: Returned FALSE or 0

Notice:

This function is supposed to be used for Artray's cameras except USB3-T2 series.

For USB3-T2 series, please call the function "ArtCam_SetExposureTime" to set up the exposure time in units of 100 ms, and call "ArtCam_GetExposureTime" to get the real exposure time in units of 100 ms.

Kinly note that the following information concerning ArtCam_SetRealExposureTime is not for USB3-T2 series:

This function is to assign the exposure time in units of seconds, and then a corresponding value will be calculated and sent to the sensor.
(Refer to the Notice of ArtCam_GetRealExposureTime for more details about the formula.)

Therefore, even though the exposure time is shown in microsecond, the accuracy in microsecond cannot be guaranteed. (The nearest value to the calculated value will be sent to the sensor.)

Please set the value within permissible range or the function will fail and ARTCAMSDK_PARAM error will be returned.

Kindly note that depending on DLL version, this function might not be applicable to some cameras.

ArtCam_GetExposureTimeEx

```
double  ArtCam_GetRealExposureTime(HACAM hACam, LPBOOL Error)
```

Function: Get the real exposure time

Argument:

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Error information

Function Detail:

This function is to get double value of exposure time in units of seconds.

Success: TRUE on Error.
Failure: FALSE on Error.
If this function is not applicable to the camera, the returned value will be -1.0.

ArtCam_SetExposureTimeEx

```
BOOL ArtCam_SetExposureTimeEx(HACAM hACam, double Value)
```

Function: Set the real exposure time

Argument:

HACAM	*hACam*	Handle for camera identification
double	*Value*	Value to be set

Function Detail:

This function is to set exposure time in units of seconds.

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

Notice:

This function is to assign the exposure time in units of seconds, and then a corresponding value will be calculated and sent to the sensor.
(The nearest value to the calculated value will be sent to the sensor.)

Please refer to dExposureExMin and dExposureExMax in CAMERAINFO to find the setting range.

ArtCam_LoadConfigFile

```
BOOL ArtCam_LoadConfigFile(HACAM hACam, LPCTSTR szFileName)
```

Function: Read defect pixel compensation

Argument:

HACAM	*hACam*	Handle for camera identification
LPCTSTR	*szFileName*	Path to the defect pixel compensation file

Function Detail:

To read file for defect pixel compensation.

After the compensation file is read, please use ArtCam_SetConfigFilter to switch on the compensation.

Success: Returned TRUE or 1

Failure: Returned FALSE or 0

Notice:

This function is available only on models with defect pixel compensation.

Defect pixel compensation file is located in the CD-rom provided with the camera. The file name is “Config_XXXXX.dat”. (XXXXX=serial number)

For more details, please contact our sales personnel.

ArtCam_SetConfigFilter

```
BOOL ArtCam_SetConfigFilter(HACAM hACam, LONG Value)
```

Function: Enable/disable defect pixel compensation

Argument:

HACAM	*hACam*	Handle for camera identification
LONG	*Value*	Set defect pixel compensation

Function Detail:

To enable/disable the defect pixel compensation.
The setting value:
Value = 1: enable
Value = 0: disable

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

Notice:

This function is available only on models with defect pixel compensation.

ArtCam_GetConfigFilter

```
LONG ArtCam_GetConfigFilter(HACAM hACam)
```

Function: Get the setting value of defect pixel compensation

Argument:

HACAM	*hACam*	Handle for camera identification

Function Detail:

To get setting value of defect pixel compensation (1 or 0).
Success: Returned current setting of defect pixel compensation

Notice:

This function is available only on models with defect pixel compensation.

ArtCam_SetInternalCorrection

```
BOOL ArtCam_SetInternalCorrection(HACAM hACam, BOOL Enable)
```

Function: Enable/disable internal correction

Argument:

HACAM	*hACam*	Handle for camera identification
BOOL	*Enable*	Set internal correction

Function Detail:

To enable/disable internal correction.

Success: Returned TRUE or 1

Failure: Returned FALSE or 0

Notice:

This function is available only on models with internal correction.

ArtCam_GetInternalCorrection

```
BOOL ArtCam_GetInternalCorrection(HACAM hACam, LPBOOL Error)
```

Function: Get the setting value of internal correction

Argument:

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Error information

Function Detail:

To get setting value of internal correction (1 or 0).

Success: Returned current setting of internal correction

Notice:

This function is available only on models with internal correction.

ArtCam_UpdateMaskData

```
BOOL ArtCam_UpdateMaskData (HACAM hACam, MASKTYPE Flg)
```

Function : Update images obtained mask data.

Argument :

HACAM	*hACam*	Handle for camera identification
MASKTYPE	*Flg*	Masktype

Function detail :

Update images obtained mask data that SDK manages internally and overwrite the last captured image.
During process of ArtCam_Preview or ArtCam_CallBackPreview, image capturing will be continuing.
When either ArtCam_SnapShot or ArtCam_Close is executed, the latest captured image will be updated.

Before capturing images, entire data of 0 will be updated if this function is called.

For masktype, choose either MASKTYPE_LOW or MASKTYPE_HIGH.
Basically, while sensor is shielded, update MASKTYPE_LOW.
And while sensor is exposed, update the MASKTYPE_HIGH.

This function is available only under full resolution.
There will be error under ROI mode.

Return value :

Success : Returned TRUE or 1
Failure : Returned FALSE or 0

ArtCam_SaveMaskFile

```
BOOL ArtCam_SaveMaskFile (HACAM hACam, LPCTSTR szFileName)
```

Function : Save current mask data.

Argument :

HACAM	*hACam*	Handle for camera identification
LPCTSTR	*szFileName*	Name of Saved file

Function detail :

Mask data managed inside SDK can be save in any file.

Return value :

Success: Returned TRUE or 1

Failure: Returned FALSE or 0

ArtCam_LoadMaskFile

```
BOOL ArtCam_LoadMaskFile (HACAM hACam, LPCTSTR szFileName)
```

Function : Load mask data from file.

Argument :

HACAM	*hACam*	Handle for camera identification
LPCTSTR	*szFileName*	File name of loading mask data

Function detail :

Load specific mask filter on SDK.

Please use the mask data provided by Artray or data saved by using ArtCam_SaveMaskFile function.

If you choose incorrect file format, an error will occur.

Return value :

Success: Returned TRUE or 1

Failure: Returned FALSE or 0

ArtCam_SetMaskFilter

```
BOOL ArtCam_SetMaskFilter (HACAM hACam, LONG Value)
```

Function : Set up valid/invalid action on mask filter process

Argument :

HACAM	*hACam*	Handle for camera identification
LONG	*Value*	Setting of mask filter

Function detail :

Set up valid/invalid action on mask filter process of output images.
By using Value, either 0 or 1 can be set.
When it is set 1, it becomes valid; when it is set 0, it becomes invalid.

ArtCam_SetFilterValue is also included in this function, so the same action will be performed when filter type, ARTCAM_FILTERTYPE_MASKFILTER in ArtCam_SetFilterValue, is chosen.

Return value :

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_GetMaskFilter

```
LONG ArtCam_Get MaskFilter(HACAM hACam, LPBOOL Error)
```

Function : Obtain current value of mask filter process

Argument :

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Pointer to the Bool variable that will receive error

Function detail :

Obtain current value of mask filter process setting.

ArtCam_GetFilterValue is also included in this function, so the same action will be performed when filter type, ARTCAM_FILTERTYPE_MASKFILTER in ArtCam_GetFilterValue, is chosen.

Return value :

Success: Returned TRUE or 1, and returned current setting of mask filter.
Failure: Returned FALSE or 0, and returned 0.

ArtCam_SetDotFilter

```
BOOL ArtCam_SetDotFilter (HACAM hACam, LONG Value)
```

Function : Set up valid/invalid action on pixel correction filter

Argument :

HACAM	*hACam*	Handle for camera identification
LONG	*Value*	Setting of pixel correction filter

Function detail :

Set up valid/invalid action on pixel correction filter of output image.

By Value, it can be set either 0 or 1.
When it is set as 1, it is valid.
On the other hand, it will be invalid when it is set as 0.

ArtCam_SetFilterValue is also included in this function, so the same action will be performed when filter type, ARTCAM_FILTERTYPE_DOTFILTER in ArtCam_SetFilterValue, is chosen.

Return value :

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

ArtCam_GetDotFilter

```
LONG ArtCam_GetDotFilter(HACAM hACam,  LPBOOL Error)
```

Function : Obtain current setting of pixel correction filter process

Argument :

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Pointer to the Bool variable that will receive error

Function detail :

Obtain current setting of pixel correction filter process.

ArtCam_GetFilterValue is also included in this function, so the same action will be performed when filter type, ARTCAM_FILTERTYPE_DOTFILTER in ArtCam_GetFilterValue, is chosen.

Return value :

Success: Returned TRUE or 1 and returned current setting of pixel correction filter
Failure: Returned FALSE or 0 and returned 0.

ArtCam_SetPeltier

```
BOOL ArtCam_SetPeltier (HACAM hACam, LONG Value)
```

Function : Control the Peltier.

Argument :

HACAM	*hACam*	Handle for camera identification
LONG	*Value*	Peltier value

Function detail :

TNIR series:
Control the pulse width of voltage caused by Peltier element inside the sensor.
By using Value, 0~127 can be set.
When it is set as 0, pulse width becomes 0%; when it is set as 127, the pulse width becomes 100%.

Other series:
To enable/disable Peltier.
The setting value:
Value = 1: enable
Value = 0: disable

ArtCam_SetFilterValue is also included in this function, so the same action will be performed when filter type, ARTCAM_FILTERTYPE_PELTIER in ArtCam_SetFilterValue , is chosen.

Return value :

Success: Returned TRUE or 1
Failure: Returned FALSE or 0

Notice :

TNIR series:
Voltage caused by Peltier element is input by AC adaptor, which is usually supplied 12V.
When the pulse width is 100%, roughly 1.5A current flows. And temperature difference between inside sensor and external package will be around 70C.
If Peltier value is set more than defaulted value 80, please design a system for dispersion of heat from the camera case.

ArtCam_GetPeltier

```
LONG ArtCam_Get Peltier(HACAM hACam,  LPBOOL Error)
```

Function : Obtain the current setting of Peltier control

Argument :

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Pointer to the Bool variable that will receive error

Function detail :

Obtain the current setting of Peltier control.

ArtCam_GetFilterValue is also included in this function, so the same action will be performed when filter type, ARTCAM_FILTERTYPE_PELTIER in ArtCam_GetFilterValue is chosen.

Return value :

Success: Returned TRUE or 1 in Error and returned Current setting of Peltier control.

Failure: Returned FALSE or 0 in Error and returned 0.

ArtCam_GetTemperature

```
LONG ArtCam_GetTemperature (HACAM hACam, LPBOOL Error)
```

Function : Obtain the sensor's voltage from internal IngaAs sensor

Argument :

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Pointer to the Bool variable that will receive error

Function detail :

This function is only available on ARTCAM-008TNIR, ARTCAM-031TNIR and ARTCAM-0016TNIR.
Obtain the sensor's voltage from internal IngaAs sensor as mV unit.
When the Peltier controlled pulse is set as 0% or 100%, a difference around 100mV will be occurred.
When it obtains higher voltage, internal temperature of InGaAs sensor will become lower.

ArtCam_GetFilterValue is also included in this function, so the same action will be performed when filter type, ARTCAM_FILTERTYPE_TEMPERATURE in ArtCam_GetFilterValue is chosen.

Return value :

Success: Returned TRUE or 1 in Error and returned status of capturing image.
Failure: Returned FALSE or 0 in Error and returned 0.

Notice :

There is 100mV of difference may occur from using 1 unit of sensor.
Please use voltage obtained from this function as a standard of internal temperature controlled by Peltier.

ArtCam_GetTemperatureEx

```
double ArtCam_GetTemperatureEx (HACAM hACam, LPBOOL Error)
```

Function : Get temperature value from thermometer built in the InGaAs sensor

Argument :

HACAM	*hACam*	Handle for camera identification
LPBOOL	*Error*	Pointer to the Bool variable that will receive error

Function detail :

Get temperature value (°C) from thermometer built in the InGaAs sensor.

Return value :

Success: Returned TRUE or 1 in Error and returned value of temperature.

Failure: Returned FALSE or 0 in Error and returned 0.

Image Filter Setting Possible Value

Basic settings

ARTCAM_FILTERTYPE_BRIGHTNESS	Set Brightness Min: -255 / Max:255 / Default: 0 Setter:ArtCam_SetBrightness Getter: ArtCam_GetBrightness
ARTCAM_FILTERTYPE_CONTRAST	Set Contrast Min: -127 / Max:127 / Default: 0 Setter:ArtCam_SetContrast Getter: ArtCam_GetContrast
ARTCAM_FILTERTYPE_SHARPNESS	Set Sharpness Min: 0 / Max:30 / Default: 0 Setter:ArtCam_SetSharpness Getter: ArtCam_GetSharpness *Frame rate might be down because it would use more CPU performance.
ARTCAM_FILTERTYPE_GAMMA	Set Gamma Value (gamma1.0=100) Min: 0 / Max:200 / Default: 100 Setter:ArtCam_SetGamma Getter: ArtCam_GetGamma *Frame rate might be down because it would use more CPU performance.
ARTCAM_FILTERTYPE_GLOBAL_GAIN	Set Global Gain Setter:ArtCam_SetGlobalGain Getter: ArtCam_GetGlobalGain *Setting value and default value is different at each sensor
ARTCAM_FILTERTYPE_EXPOSURETIME	Set Shutter Speed Setter:ArtCam_SetExposureTime Getter: ArtCam_GetExposureTime *Setting value and default value is different at each sensor

	Global Gain			Shutter Speed			Peltier
	Min	Max	Default	Min	Max	Default	
ARTCAM-008TNIR	1	1	1	1	32767	100	N/A
ARTCAM-031TNIR	0	1	1	0	65500	100	Available
ARTCAM-0016TNIR	N/A			5	65535	10000	N/A

* About Global Gain

ARTCAM-031TNIR can switch Lo Gain and Hi Gain.

When it is set 1, it becomes Hi Gain; when it is set 0, it becomes Lo Gain.

Grayscale Filter Setting Possible Value

ARTCAM_FILTERTYPE_GRAY_MODE			
Set Grayscale Mode	Min: 0	Max: 2	Def: 0
Setter: ArtCam_SetGrayMode	Getter: ArtCam_GetGrayMode		

0 = GRAY_NONE // Invalid Still Bayer arrangement
1 = GRAY_BAYERCONVERT // To Bayer Arranement Add calculation by GRAY GAIN and GRAY OFFSET
2 = GRAY_GRAYSCALE // After change color, leave anly luminance information.			

ARTCAM_FILTERTYPE_GRAY_GAIN_R / GAIN_G1 / GAIN_G2 / GAIN_B			
Control each color's gain by a software	Min: 0	Max: 400	Def: 128
Setter: ArtCam_SetGrayGainRed	Getter: ArtCam_GetGrayGainRed		
Setter: ArtCam_SetGrayGainGreen1	Getter: ArtCam_GetGrayGainGreen1		
Setter: ArtCam_SetGrayGainGreen2	Getter: ArtCam_GetGrayGainGreen2		
Setter: ArtCam_SetGrayGainBlue	Getter: ArtCam_GetGrayGainBlue		

ARTCAM_FILTERTYPE_GRAY_OFFSET_R / OFFSET_G1 / OFFSET_G2 / OFFSET_B			
Control each color's offset by a software	Min: -255	Max: 255	Def: 0
Setter: ArtCam_SetGrayOffsetRed	Getter: ArtCam_GetGrayOffsetRed		
Setter: ArtCam_SetGrayOffsetGreen1	Getter: ArtCam_GetGrayOffsetGreen1		
Setter: ArtCam_SetGrayOffsetGreen2	Getter: ArtCam_GetGrayOffsetGreen2		
Setter: ArtCam_SetGrayOffsetBlue	Getter: ArtCam_GetGrayOffsetBlue		

****
****

ARTRAY Camera / Capture Module Software Developer Kit
Dynamic Link Library for Windows XP,Vista,7,8,10

ARTRAY CO., LTD.
4F Ueno Bldg,1-17-5 Kouenjikita,Suginami-ku, Tokyo 166-0002 Japan
TEL: 03-3389-5488
FAX: 03-3389-5486
E-mail: sales@artray.us
URL: www.artray.us