

2008-10-07版

ARTRAY Camera / Capture Module Software Developer Kit

---Functions Manual---

Dynamic Link Library for Windows2000,XP

2008.10.07

株式会社アートレイ

■DLL 初期化関連■

ArtCam_GetDllVersion	P4
ArtCam_GetLastError	P5
ArtCam_Initialize	P6
ArtCam_Release	P6

■映像取り込み関連■

ArtCam_Preview	P7
ArtCam_Record	P7
ArtCam_CallBackPreview	P8
ArtCam_SnapShot	P9
ArtCam_Capture	P10
ArtCam_Close	P10
ArtCam_Trigger	P11
WM_GRAPHPAINT	P12
WM_ERROR	P13
ArtCam_StartPreview	P14
ArtCam_StopPreview	P14
ArtCam_SaveImage	P15
ArtCam_GetImage	P16

■設定ダイアログ■

ArtCam_SetCameraDlg	P17
ArtCam_SetImageDlg	P17
ArtCam_SetAnalogDlg	P18

■カメラ設定関連■

ArtCam_SetPreviewWindow	P19
ArtCam_SetCaptureWindow	P20
ArtCam_SetCaptureWindowEx	P21
ArtCam_GetCaptureWindowEx	P22
ArtCam_SetColorMode	P23
ArtCam_GetColorMode	P24
ArtCam_SetCrossbar	P24
ArtCam_SetDeviceNumber	P25
ArtCam_GetDeviceName	P25
ArtCam_EnumDevice	P26
ArtCam_SetCameraType	P27
ArtCam_GetCameraType	P27
ArtCam_Width	P28
ArtCam_Height	P28
ArtCam_Fps	P29
ArtCam_GetCameraInfo	P29
ArtCam_SetIOPort	P30
ArtCam_GetIOPort	P30
ArtCam_SetSubSample	P31

ArtCam_GetSubSample	P31
ArtCam_SetWaitTime	P32
ArtCam_GetWaitTime	P32
ArtCam_SetMirrorV	P33
ArtCam_GetMirrorV	P33
ArtCam_SetMirrorH	P34
ArtCam_GetMirrorH	P34
ArtCam_SetHalfClock	P35
ArtCam_GetHalfClock	P35
ArtCam_SetAutoIris	P36
ArtCam_GetAutoIris	P36
ArtCam_SetSamplingRate	P37
ArtCam_GetSamplingRate	P37
ArtCam_GetVideoFormat	P38
ArtCam_WriteSromID	P39
ArtCam_ReadSromID	P39
ArtCam_WriteRegister	P40
ArtCam_ReadSRegister	P40
ArtCam_SetFilterValue	P41
ArtCam_GetFilterValue	P41
ArtCam_Set***	P41
ArtCam_Get***	P41
ArtCam_SetSyncV	P43
ArtCam_GetSyncV	P43
ArtCam_SetSyncH	P44
ArtCam_GetSyncH	P44

■ 画像フィルター設定可能値 ■

全カメラ共通	P45
グレースケール変換	P46
ARTCNVII	P47

■ArtCam_GetDllVersion

定義: **DWORD** ArtCam_GetDllVersion(*void*)

機能: ライブラリのバージョンを取得する

引数: なし

機能詳細:

DLL のバージョンと、DLL の種類を取得します。
返される DWORD(32bit)値には、上位16ビットに DLL タイプ、
下位16ビットに DLL のバージョンが格納されます。
ライブラリを使う際に、インストールされてある DLL のバージョンと、
SDK のバージョンをチェックするのが望ましいです。

バージョンは4桁の整数で取得されます。
バージョンが 1.278 の場合下位 16 ビットには 1278 が格納されます。

DLL タイプは以下の通りです

コード	デバイスタイプ
ARTCAM_CAMERATYPE_DS	DirectShowCamera
ARTCAM_CAMERATYPE_USTC	ARTUST
ARTCAM_CAMERATYPE_CNV	ARTCNV
ARTCAM_CAMERATYPE_130MI	ARTCAM-130MI
ARTCAM_CAMERATYPE_200MI	ARTCAM-200MI
ARTCAM_CAMERATYPE_300MI	ARTCAM-300MI
ARTCAM_CAMERATYPE_150P	ARTCAM-150P
ARTCAM_CAMERATYPE_320P	ARTCAM-320P
ARTCAM_CAMERATYPE_200SH	ARTCAM-200SH
ARTCAM_CAMERATYPE_098	ARTCAM-098
ARTCAM_CAMERATYPE_036MI	ARTCAM-036MI
ARTCAM_CAMERATYPE_500P	ARTCAM-500P
ARTCAM_CAMERATYPE_150P2	ARTCAM-150P II
ARTCAM_CAMERATYPE_036MIST	ARTCAM-036MI-TWIN
ARTCAM_CAMERATYPE_500MI	ARTCAM-500MI
ARTCAM_CAMERATYPE_T111	ARTCAM-T111
ARTCAM_CAMERATYPE_T112	ARTCAM-T112
ARTCAM_CAMERATYPE_150P3	ARTCAM-150P III
ARTCAM_CAMERATYPE_130MI_MOUT	ARTCAM-130MI-MOUT
ARTCAM_CAMERATYPE_150P3_MOUT	ARTCAM-150P III-MOUT
ARTCAM_CAMERATYPE_267KY	ARTCAM-267KY
ARTCAM_CAMERATYPE_274KY	ARTCAM-274KY
ARTCAM_CAMERATYPE_SATA	SATA カメラ

■ArtCam_GetLastError

定義: **LONG** ArtCam_GetLastError(HACAM *hACam*)

機能: エラーを取得する

引数:

HACAM *hACam* カメラ識別用ハンドル

機能詳細:

関数の戻り値でエラーが返った際にこの関数を呼ぶと
エラーコードを取得する事が出来ます。
エラーコードとエラー内容の詳細については下表を参照してください。

エラーはスタック型のデータ構造で格納されていて、
後に発生したエラーから順番に一つずつ呼び出すことが出来ます。

エラーコード	エラー詳細
ARTCAMSDK_NOERROR	正常
ARTCAMSDK_NOT_INITIALIZE,	初期化されてません
ARTCAMSDK_DISABLEDDEVICE,	利用不可能なデバイスにアクセスしようとした
ARTCAMSDK_CREATETHREAD,	画像取り込み用スレッド作成に失敗
ARTCAMSDK_CREATEWINDOW,	ウィンドウ作成に失敗
ARTCAMSDK_OUTOFMEMORY,	イメージ転送に十分なメモリを確保していない 又はメモリの確保に失敗
ARTCAMSDK_CAMERASET,	カメラ(デバイス)の設定でエラー
ARTCAMSDK_CAMERASIZE,	カメラ(デバイス)のサイズ設定でエラー
ARTCAMSDK_CAPTURE,	映像取り込みで失敗
ARTCAMSDK_PARAM,	引数が間違ってます
ARTCAMSDK_DIRECTSHOW,	DirectShow 初期化エラー
ARTCAMSDK_UNSUPPORTED,	この機能はサポートされていません
ARTCAMSDK_UNKNOWN,	不明のエラー
ARTCAMSDK_CAPTURELOST,	デバイスが消失
ARTCAMSDK_FILENOTFOUND,	指定ファイルが見つからない
ARTCAMSDK_FPGASET,	FPGA の設定でエラー
ARTCAMSDK_TRANSIMAGEFAILED	画像転送エラー

■ArtCam Initialize

定義: **HACAM** ArtCam_Initialize(HWND *hWnd*)

機能: ライブラリを初期化する

引数:

HWND	<i>hWnd</i>	メッセージ取得用ウィンドウハンドル
------	-------------	-------------------

機能詳細:

関ライブラリの初期化を行います。
本ライブラリを使用する際は初めにこの関数を呼び出してください。

本関数成功時にはカメラ識別用ハンドルが戻り値に得られます。
本関数に失敗した場合、NULLもしくは 0 が返ります。

*hWnd*にウィンドウハンドルを設定すると、エラーが発生した際に
WM_ERRORがウィンドウプロシージャに対して送られます。

また、この関数が呼ばれるときに、前回終了時の各パラメータの設定を
レジストリから読み込みます。
各パラメータの設定は下記のレジストリキーに保存されています。
(保存されないパラメータもありますのでご注意ください)

HKEY_CURRENT_USER¥Software¥Artray¥ArtCam[機種名]Sdk

■ArtCam Release

定義: **BOOL** ArtCam_Release(HACAM *hACam*)

機能: ライブラリを開放する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

接続しているカメラを開放し、内部データも全て初期化されます
アプリケーション終了時か、カメラの使用を終了する際に呼び出してください。

成功時には **TRUE** もしくは 1 が返ります。
失敗時には **FALSE** もしくは 0 が返ります。

この関数を呼び出した後で再度カメラの表示を行いたい場合は
もう一度 ArtCam Initialize を呼び出してください。

また、この関数が呼ばれるときに、各パラメータ設定をレジストリに保存します。
(保存されないパラメータもありますのでご注意ください)

■ArtCam Preview

定義: **BOOL** ArtCam_Preview(HACAM *hACam*)

機能: 映像の表示

引数:

HACAM *hACam* カメラ識別用ハンドル

機能詳細:

SDK側で制御してカメラの映像を表示します。
この関数を使用する前には必ず ArtCam Initialize を呼び出してください。
この関数が成功した場合は ArtCam SetPreviewWindow で指定した
ウィンドウ領域に子ウィンドウを作成してそのウィンドウに映像を描画します。
ArtCam SetPreviewWindow で設定されていない場合は
新しいウィンドウが作成されて、そこに映像が描画されます。

関数が成功したときは TRUE 若しくは 1 が、
失敗したときには FALSE 若しくは 0 が返ります。

■ArtCam Record

定義: **BOOL** ArtCam_Record(

HACAM *hACam*, LPCTSTR *lpAviName*, UINT *RecTime*, BOOL *fShow*)

機能: ファイルへ録画

引数:

HACAM	<u><i>hACam</i></u>	カメラ識別用ハンドル
LPCTSTR	<u><i>lpAviName</i></u>	保存するファイル名
UINT	<u><i>RecTime</i></u>	録画時間(ミリ秒)0で連続録画
BOOL	<u><i>fShow</i></u>	映像を表示するかどうか

機能詳細:

RecTime を指定した場合、指定時間を経過すると自動で録画は終了しますが、
デバイスは開放されませんので、カメラの映像(表示している場合)は
そのまま表示され続けます。
録画終了と同時になんらかの処理が行いたい場合はタイマー等を使用して
録画終了のタイミングを取得する必要があります。
fShow に関しては映像を表示しない方がフレーム落ちなどは起こりにくくなります。

関数が成功したときは TRUE 若しくは 1 が、
失敗したときには FALSE 若しくは 0 が返ります。

本関数は **ArtCamSdk.dll** (DirectShow カメラ)専用です。

■ArtCam_CallBackPreview

定義: BOOL ArtCam_CallBackPreview(

HACAM *hACam*, HWND *hWnd*, LPBYTE *lpImage*, LONG *Size*, BOOL *TopDown*)

機能: 映像データを取得しながら動画表示

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
HWND	<i>hWnd</i>	メッセージを受け取るウィンドウハンドル
LPBYTE	<i>lpImage</i>	映像のデータを受け取る配列のアドレス
LONG	<i>Size</i>	<i>lpImage</i> の配列長
BOOL	<i>TopDown</i>	画像の上下判定

機能詳細:

hWnd にウィンドウハンドルを指定した場合、
指定したウィンドウプロシージャに WM_GRAPHPAINT が送られてきます。

lpImage と *Size* を指定している場合は WM_GRAPHPAINT の前に *lpImage* で指定した配列にカメラの映像がコピーされます。

この場合、カメラの映像のサイズと同じかそれよりも大きいサイズの配列を用意しておかないと映像はコピーされません。また、*lpImage* に一時的な配列のアドレスを入れてはいけません。

用意したビットマップが DDB(トップダウン) ならば *TopDown* に TRUE を、
DIB(ボトムアップ) ならば *TopDown* に FALSE を設定してください。

なお、本関数でも ArtCam_Preview 同様自動描画機能があります。
自動描画の方法は ArtCam_Preview と同じです。

関数が成功したときは TRUE 若しくは 1 が、
失敗したときには FALSE 若しくは 0 が返ります。

注意事項:

VB ではメッセージを取得することが比較的困難な上、
VB の処理速度の関係上不具合が出ることもあります。
関数自体は使えますが、WM_GRAPHPAINT によるリアルタイム処理は
控えた方が良いでしょう。
(現在のサンプルではその方法は廃止してタイマーで描画のタイミングを制御しています)

■ArtCam SnapShot

定義: **BOOL** ArtCam_SnapShot(

HACAM *hACam*, LPBYTE *lpImage*, LONG *Size*, BOOL *TopDown*)

機能: カメラの映像を1回だけ取得します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBYTE	<i>lpImage</i>	映像のデータを受け取る配列のアドレス
LONG	<i>Size</i>	<i>lpImage</i> の配列長
BOOL	<i>TopDown</i>	画像の上下判定

機能詳細:

ソフトトリガーによりカメラの映像を 1 枚だけ取得します。
関数が成功した場合 *lpImage* に取得した映像のデータが格納されます。
この際、カメラの映像のサイズと同じかそれよりも大きいサイズの配列を
用意しておかないと映像は取得できません。

用意したビットマップが DDB(トップダウン)ならば *TopDown* に TRUE を
DIB(ボトムアップ)ならば *TopDown* に FALSE を設定してください。

ArtCam_GetImage との違いは ArtCam_GetImage が動画表示中に1ショット分の
画像を取得するのに対して ArtCam_SnapShot は映像を表示していない状態で
1ショット分の画像を取得する点です。
本関数は ArtCam_Preview など動画が表示されている場合失敗します。

関数が成功したときは TRUE 若しくは 1 が、
失敗したときには FALSE 若しくは 0 が返ります。

注意事項:

一部のストロボ対応カメラ(ARTCAM-300MI-STR2 など)でこの関数を呼ぶと、
センサ側からカメラの BNC 端子にストロボ信号が送られます。
信号のタイミングなどの詳細は商品説明書を参照してください。
標準のカメラにはこの機能はございません。

■ArtCam_Capture

定義: **BOOL** ArtCam_Capture(HACAM hACam)

機能: 連続スナップショット用にカメラを初期化します

引数:

HACAM *hACam* カメラ識別用ハンドル

機能詳細:

ArtCam_SnapShot を連続で使用する為にカメラを初期化します。
通常 ArtCam_SnapShot を使用した場合は初期化→取得→開放という手順をしますが、
本関数であらかじめ初期化しておくことで ArtCam_SnapShot を呼んだ際の
初期化と開放の処理が無視され、高速に映像を取得できます。
ArtCam_SnapShot の使用をやめてカメラを開放する場合は ArtCam_Close を呼びます。

大まかな流れとしては

ArtCam_Capture (初期化)
↓
ArtCam_SnapShot (連続で使用可能)
↓
ArtCam_Close (開放)

このようになります。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_Close

定義: **BOOL** ArtCam_Close(HACAM *hACam*)

機能: デバイスを開放する

引数:

HACAM *hACam* カメラ識別用ハンドル

機能詳細:

カメラの表示を中止して、デバイスを開放します。
ArtCam_Preview
ArtCam_Record
ArtCam_CallbackPreview
ArtCam_Capture
ArtCam_Trigger
にて映像の取得を開始した場合は、この関数を使用して開放してください。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam Trigger

定義: **BOOL** ArtCam_Trigger(HACAM hACam, HWND hWnd, LPBYTE lpImage, LONG Size, **BOOL** TopDown)

機能: 外部トリガーモードでカメラ映像を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
HWND	<i>hWnd</i>	メッセージを受け取るウィンドウハンドル
LPBYTE	<i>lpImage</i>	映像のデータを受け取る配列のアドレス
LONG	<i>Size</i>	lpImage の配列長
BOOL	<i>TopDown</i>	画像の上下判定

機能詳細:

本関数の使用方法は ArtCam_CallBackPreview と基本的に同じです。
ArtCam_CallBackPreview との違いは
映像が入ってくるタイミングがカメラの動作クロックに依存するか
外部トリガーが発せられた際かという点です。

本関数でカメラを初期化した場合は、トリガーがカメラに送られた時のみ
映像の更新&取得メッセージが送られてきます。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

注意事項:

トリガ信号は、**0-5V** ～**0-12V** のパルス信号を使用してください。
トリガが入力されてから撮影までのタイミングはカメラの機種によって異なります。
詳細については商品説明書を参照してください。

■ WM_GRAPHPAINT

定義: `#define WM_GRAPHPAINT WM_APP + 2`

機能: カメラ映像が更新されたときに発行されるメッセージ

WPARAM	wParam	LPGP_INFO
LPARAM	lParam	常に NULL

機能詳細:

wParam で受け取れる LPGP_INFO は、映像の情報を格納している構造体へのポインタです。

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

LPGP_INFO lpGPIF = (LPGP_INFO)wParam

ArtCam CallBackPreview、ArtCam Trigger で

hWnd にウィンドウハンドルを設定した場合、
設定したウィンドウのコールバックプロシージャに対してこのメッセージが送られてきます。
このメッセージはカメラの映像が更新されたタイミングで送られてきます。

ArtCam CallBackPreview で映像のデータを取得したい場合は、
引数の *lpImage*、*Size* に配列のポインタと配列長を指定しておけば、
メッセージを受け取った時点で指定した配列に最新の画像データが格納されます。

wParam が NULL の場合、WM_GRAPHPAINT はエラー情報となり、
その場合の *wParam*、*lParam* の意味は WM_ERROR と同様になります。

なお、WM_GRAPHPAINT は 0x8002 と定義されています。

■WM_ERROR

定義: `#define WM_GRAPHPAINT WM_APP + 3`

機能: エラーが発生したときに発行されるメッセージ

WPARAM	wParam	常に 0
LPARAM	lParam	エラーコード

機能詳細:

ArtCam_Initialize でウィンドウハンドルを指定した場合、SDK 内で何らかのエラーが発生した際にエラーコードがウィンドウプロシージャに送られます。

WM_ERROR は 0x8003 と定義されています。

エラーコードは以下の通りです。

エラーコード	ステータス
ARTCAMSDK_NOERROR	正常
ARTCAMSDK_NOT_INITIALIZE	初期化されていません
ARTCAMSDK_DISABLEDDEVICE	利用不可能なデバイスにアクセスしようとした
ARTCAMSDK_CREATETHREAD	画像取り込み用スレッド作成に失敗
ARTCAMSDK_CREATEWINDOW	ウィンドウ作成に失敗
ARTCAMSDK_OUTOFMEMORY	イメージ転送に十分なメモリを確保していない 又は メモリの確保に失敗
ARTCAMSDK_CAMERASET	カメラ(デバイス)の設定でエラー
ARTCAMSDK_CAPTURE	映像取り込みで失敗
ARTCAMSDK_PARAM	引数が間違っています
ARTCAMSDK_DIRECTSHOW	DirectShow 初期化エラー
ARTCAMSDK_UNSUPPORTED	この機能はサポートされていません
ARTCAMSDK_UNKNOWN	不明のエラー
ARTCAMSDK_CAPTURELOST	デバイスが消失
ARTCAMSDK_FILENOTFOUND	指定ファイルが見つからない
ARTCAMSDK_FPGASET	FPGA の設定でエラー

■ArtCam_StartPreview

定義: **BOOL** ArtCam_StartPreview(HACAM *hACam*)

機能: カメラの表示を開始します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

カメラの描画を開始します。

この関数は

ArtCam_Preview

ArtCam_Record

ArtCam_CallBackPreview

では内部的に使用されています。

SDK 利用者がこの関数を使う場合は ArtCam_StopPreview を呼び出してカメラの表示を一時的に停止したものを再び表示させる時だけです。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_StopPreview

定義: **BOOL** ArtCam_StopPreview(HACAM *hACam*)

機能: カメラの表示を停止します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

カメラ映像の表示を中止します。

この関数ではデバイスは開放されないので、

一時的に表示を中断したい時のみ使用してください。

再び表示させる時は ArtCam_StartPreview を使用します

この関数では ArtCam_Preview、ArtCam_CallBackPreview で画像を表示している時のみ使用可能です。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_SaveImage

定義: **BOOL** ArtCam_SaveImage(

HACAM *hACam*, LPCTSTR *lpSaveName*, LONG *FileType*)

機能: カメラの映像を保存する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPCTSTR	<i>lpSaveName</i>	保存ファイル名
LONG	<i>FileType</i>	保存タイプ

機能詳細:

カメラの映像をファイルに保存します。

保存される映像は、ArtCam_Preview、ArtCam_CallBackPreview、ArtCam_SnapShot、ArtCam_Trigger 等の映像取得関数で取得した最後の映像です。

ただし、ArtCam_Preview、ArtCam_CallBackPreview で

映像をリアルタイムで取得している際に保存すると

環境やカメラの動作クロック、保存形式等によっては保存する映像が

崩れる場合があります。ご注意ください。

映像が崩れる際は ArtCam_StopPreview で映像の更新を一時的に止めると崩れることがなくなります。

関数が成功したときは **TRUE** 若しくは **1** が、

失敗したときには **FALSE** 若しくは **0** が返ります。

注意事項:

本関数はカメラから取得した生の映像を保存する為の関数です。

アプリケーション側で何らかの加工を行った映像を保存する場合は

アプリケーション側にて独自に保存してください。

保存形式はビットマップ、バイナリ(RAW)、JPEG(高品質、標準、低品質)、PNG、TIFF から選べます。

本関数にて保存できるファイルの扱いに関してですが、BMP、RAW の

読み込み方法のみサポート対象とさせていただきます。

それ以外の形式の読み込み方法、BMP、RAW の保存方法などに関しては

一切お答えできませんので予めご了承ください。

※16ビット(10ビット)転送モードの際の画像保存に関して

JPEG では16ビット画像を保存する事はできません。

JPEG で保存する場合はビット深度 8ビットのグレースケールとして保存されます。

BMP、RAW では16ビットでも保存できますが、パレットの情報などはファイルに

保存しておりませんので、専用読みこみルーチンを組んだソフト以外では正しく表示できないことがあります。

16ビット画像を保存する場合は PNG、TIFF を使用することをお勧めします。

この形式に関しては弊社製ビューアソフトウェア、Adobe 製 Photoshop6にて

読み込めることを確認しています。

ただし、全ての画像ソフトウェアが16ビット画像に対応している訳では

ありませんので、保存後に使用する際は注意してください。

■ArtCam_GetImage

定義: **BOOL** ArtCam_GetImage(

HACAM *hACam*, LPBYTE *lpImage*, LONG *Size*, BOOL *TopDown*)

機能: カメラの表示を取得します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBYTE	<i>lpImage</i>	映像のデータを受け取る配列のアドレス
LONG	<i>Size</i>	<i>lpImage</i> の配列長
BOOL	<i>TopDown</i>	画像の上下判定

機能詳細:

カメラの映像を取得します。

関数が成功した場合 *lpImage* に直前に取得した映像のデータが格納されます。

この際、カメラの映像のサイズと同じかそれよりも大きいサイズの配列を

用意しておかないと映像は取得できません。

用意したビットマップが DDB(トップダウン)ならば *TopDown* に真を

DIB(ボトムアップ)ならば *TopDown* に偽を設定してください。

関数が成功したときは TRUE 若しくは 1 が、

失敗したときには FALSE 若しくは 0 が返ります。

注意事項:

本関数は ArtCam_Preview もしくは ArtCam_CallBackPreview を使用している際に非同期で映像を取得する為の関数です。

1枚だけ映像を取得する場合は ArtCam_SnapShot を使用してください。

本関数は PC のスペックが低い環境や、VB 等の処理の遅い言語で作成する際に使用することを想定してあります。

それ以外の言語で作成する際は WM_GRAPHPAINT で映像更新のメッセージを受け取って処理してください。

■ArtCam_SetCameraDlg

定義: **BOOL** ArtCam_SetCameraDlg(HACAM *hACam*, HWND *hWnd*)

機能: カメラ設定ダイアログを表示する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
HWND	<i>hWnd</i>	ダイアログを表示する親ウィンドウ

機能詳細:

この関数では主に映像のサイズやフレームレート、ビット深度等を設定できるダイアログボックスを表示します。
表示されるダイアログボックスは接続しているデバイスによって変わります。
映像を表示した状態でこの関数が呼ばれた際は一時的にカメラの映像は停止され、ダイアログボックスを閉じた際に表示を再開されます。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_SetImageDlg

定義: **BOOL** ArtCam_SetImageDlg(HACAM *hACam*, HWND *hWnd*)

機能: フィルター設定ダイアログを表示する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
HWND	<i>hWnd</i>	ダイアログを表示する親ウィンドウ

機能詳細:

この関数では主に映像の明るさ、コントラスト、ホワイトバランス等を設定できるダイアログボックスを表示します。
表示されるダイアログボックスは接続しているデバイスによって変わります。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_SetAnalogDlg

定義: **BOOL** ArtCam_SetCameraDlg(HACAM *hACam*, HWND *hWnd*)

機能: ポート／カメラ設定ダイアログを表示する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
HWND	<i>hWnd</i>	ダイアログを表示する親ウィンドウ

機能詳細:

この関数では主にアナログポート、カメラデバイス内設定等を設定できるダイアログボックスを表示します。
表示されるダイアログボックスは接続しているデバイスによって変わります。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_SetPreviewWindow

定義: **BOOL** ArtCam_SetPreviewWindow(

HACAM *hACam*, HWND *hWnd*, LONG *Left*, LONG *Top*, LONG *Right*, LONG *Bottom*)

機能: カメラの映像を表示するウィンドウと領域を指定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
HWND	<i>hWnd</i>	表示するウィンドウのハンドルを指定します
LONG	<i>Left</i>	表示する矩形の左上の X 軸座標を指定します
LONG	<i>Top</i>	表示する矩形の左上の Y 軸座標を指定します
LONG	<i>Right</i>	表示する矩形の右下の X 軸座標を指定します
LONG	<i>Bottom</i>	表示する矩形の右下の Y 軸座標を指定します

機能詳細:

hWnd にウィンドウハンドルを指定した場合は、そのウィンドウに子ウィンドウを作成してその子ウィンドウに対して描画を行います。

hWnd に NULL を指定した場合は新たにウィンドウを作成してそこに描画をします。

関数が成功したときは **TRUE** 若しくは **1** が、失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_SetCaptureWindow

定義: **BOOL** ArtCam_SetCaptureWindow(

HACAM *hACam*, LONG *Width*, LONG *Height*, LONG *Fps*)

機能: カメラの映像サイズとフレームレートを指定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Width</i>	カメラの幅をピクセル単位で指定します
LONG	<i>Height</i>	カメラの高さをピクセル単位で指定します
LONG	<i>Fps</i>	フレームレートを指定します。

機能詳細:

～DirectShow カメラの場合(ArtCamSdk.dll 使用時)～
Fps には秒間表示フレーム数の 10 倍の数値を指定します
秒間 30 フレームだったら 300 となります。

～それ以外の場合～
Fps は無視されます。
Width と *Height* を有効画素として既に登録されているサイズの
最も適切な数値が設定されます。

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

注意事項:

この関数は、DirectShow カメラでの録画設定に使用されることを想定しております。
それ以外のカメラの場合は、通常は ArtCam_SetCaptureWindowEx を使用してください。

Initialize にて初期化後、この関数で正しい画像サイズを設定していないと、
CallbackPreview、Snapshot 関数などに失敗する可能性があります。

■ArtCam_SetCaptureWindowEx

定義: **BOOL** ArtCam_SetCaptureWindowEx(*HACAM hACam*, **LONG** *HTotal*,
LONG *HStart*, **LONG** *HEffective*, **LONG** *VTotal*, **LONG** *VStart*, **LONG** *VEffective*)

機能: カメラの映像サイズを指定します (ROI 機能)

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>HTotal</i>	カメラの水平トータル幅をピクセル単位で指定します
LONG	<i>HStart</i>	カメラの水平開始位置を指定します
LONG	<i>HEffective</i>	カメラの水平有効幅を指定します
LONG	<i>VTotal</i>	カメラの垂直トータル幅をピクセル単位で指定します
LONG	<i>VStart</i>	カメラの垂直開始位置を指定します
LONG	<i>VEffective</i>	カメラの垂直有効幅を指定します

機能詳細:

取り込む映像のサイズを指定します。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

注意事項:

ROI 機能は、CMOS センサカメラのみの機能です。
CCD センサカメラでは映像サイズを設定することは出来ません。

カラー画像の場合、ベイヤ変換の関係で、水平、垂直ともに 5 ピクセル以上の
有効画素が必要です。

また、有効水平画素、有効垂直画素は 4 の倍数を設定することを推奨いたします。
特に、有効水平画素に 4 の倍数以外を設定した場合、
画像が正しく表示されない場合があります。

DirectShow カメラではこの機能を使うことは出来ません。
ArtCam_SetCaptureWindow で大きさを変えることは出来ます。

■ArtCam_GetCaptureWindowEx

定義: **BOOL** ArtCam_GetCaptureWindowEx(HACAM *hACam*, LONG* *HTotal*, LONG* *HStart*, LONG* *HEffective*, LONG* *VTotat*, LONG* *VStart*, LONG* *VEffective*)

機能: カメラの映像サイズを取得します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG*	<i>HTotal</i>	カメラの水平トータル幅をピクセル単位でかえします
LONG*	<i>HStart</i>	カメラの水平開始位置をかえします
LONG*	<i>HEffective</i>	カメラの水平有効幅をかえします
LONG*	<i>VTotat</i>	カメラの垂直トータル幅をピクセル単位でかえします
LONG*	<i>VStart</i>	カメラの垂直開始位置をかえします
LONG*	<i>VEffective</i>	カメラの垂直有効幅をかえします

機能詳細:

～ArtCamSdk.dll の場合～

ArtCam_Width ArtCam_Height ArtCam_Fps
をそれぞれ使用してください。

～それ以外の場合～

カメラの各種設定サイズを取得します

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam SetColorMode

定義: **BOOL** ArtCam_SetColorMode(HACAM *hACam*, LONG *ColorMode*)

機能: 取り込みのカラーモードを設定する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>ColorMode</i>	データビット数

機能詳細:

画像のビット数を指定します。

8	8ビットモノクロ画像
16	16ビットモノクロ画像 (有効ビット数は機種により異なります)
24	24ビットカラー画像 (BGR 各 8ビット)
32	32ビットカラー画像 (BGRA 各 8ビット Aは無効)
48	48ビットカラー画像 (BGR 各 16ビット)
64	64ビットカラー画像 (BGRA 各 16ビット Aは無効)

16(10)ビットに関しては環境によって若干数値が変わります。
アプリケーションを作成する際は 10、12、14 等の数値にも
対応できるように作成してください。

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

拡張機能:

[ArtCamSdk SATA.dll](#) では、引数に 10,12,14 を設定することで
有効ビット数を明示的に設定することが出来ます。
LVDS-SATA コンバータ等で、入力ビット数を変更する際に使用します。

また、16ビット転送モードの際のカラー (48bit,64bit カラー) 表示をする場合は、
下記の数字を *ColorMode* に指定してください。

	48bit カラー画像	64bit カラー画像
10bit 有効	42	58
12bit 有効	44	60
14bit 有効	46	62
16bit 有効	48	64

[ArtCam_GetColorMode](#) 関数で返される値も、上記と同様になります。

注意事項:

DirectShow カメラ(ArtCamSdk.dll)使用の場合、
映像は常に 24ビットカラーになります。
この関数での色空間の設定は出来ません。

■ArtCam_GetColorMode

定義: **LONG** ArtCam_GetColorMode (HACAM *hACam*)

機能: 現在のカラーモードを取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

この関数が成功したときは、画像のビット数が LONG 値で返されます。
関数が失敗したときには -1 が返されます。
ArtCamSdk_Sata.dll での戻り値については、前ページをご参考ください。

■ArtCam_SetCrossbar

定義: **BOOL** ArtCam_SetCrossbar(HACAM *hACam*, LONG *Input*, LONG *Output*)

機能: 接続するアナログポートを指定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Input</i>	入力ポート番号
LONG	<i>Output</i>	出力ポート番号

機能詳細:

接続するアナログ入力ポートを変更する事が出来ます。
切り替えは、*Input* に 0 からの整数を指定します。
Output は拡張機能ですので、通常は 0 を指定してください。

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

注意事項:

この関数は、ArtCnv シリーズの専用関数です。
ArtCnvII-2ch や ArtCnv-HAKO など、アナログの接続ポートを複数持つ機種において、
入力ポートを切り替える目的にのみ使用します。
入力ポートを切り替えた直後(200 ミリ秒程度)は、アナログ信号の同期を取るために
画像が乱れることがありますので、ポートを切り替えた直後の取り込みには注意してください。

■ArtCam_SetDeviceNumber

定義: **BOOL** ArtCam_SetDeviceNumber(HACAM *hACam*, LONG *Number*)

機能: 接続するデバイスの番号を指定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Number</i>	デバイスの番号を 0～9 で指定します

機能詳細:

この関数を呼び出した後にデバイスを

[ArtCam_Preview](#)

[ArtCam_Record](#)

[ArtCam_CallBackPreview](#)

[ArtCam_SnapShot](#)

[ArtCam_Capture](#)

[ArtCam_Trigger](#)

等で初期化すると指定したデバイスの映像が表示されます。

デバイスの番号を調べるには

[ArtCam_GetDeviceName](#)

[ArtCam_EnumDevice](#)

を使用してください。

関数が成功したときは TRUE 若しくは 1 が、
失敗したときには FALSE 若しくは 0 が返ります。

■ArtCam_GetDeviceName

定義: **BOOL** ArtCam_GetDeviceName(

HACAM *hACam*, LONG *index*, LPSTR *szDeviceName*, LONG *nSize*)

機能: 指定したデバイスの名前を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>index</i>	デバイスの番号を 0～9 で指定します
LPSTR	<i>szDeviceName</i>	利用可能ならばデバイスの名前がコピーされます
LONG	<i>nSize</i>	<i>szDeviceName</i> のサイズ

機能詳細:

index で指定した番号のデバイスが利用可能かチェックして、
利用可能ならばそのデバイスの名称を *szDeviceName* に格納します。

関数が成功したときは TRUE 若しくは 1 が、
失敗したときには FALSE 若しくは 0 が返ります。

■ArtCam EnumDevice

定義: **LONG** ArtCam_EnumDevice(HACAM *hACam*, TCHAR *szDeviceName*[10][256])

機能: 利用可能なデバイスの名前を列挙する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
TCHAR	<i>szDeviceName</i>	利用可能なデバイスの名前がコピーされます

機能詳細:

利用可能なデバイス名が *szDeviceName* に格納されます。
例えば、ARTCAM-130MI が 2 台利用可能な場合、通常は
szDeviceName[0] に、文字列 **ArtCam130MI_0** が格納され、
szDeviceName[1] に、文字列 **ArtCam130MI_1** が格納されます。

ArtCam_SetDeviceNumber で指示する数値は *szDeviceName* の
格納されている配列番号と同様です。

上の例で **ArtCam130MI_1** を使用したいときは、
ArtCam_SetDeviceNumber(1) のように指定してください。

また、関数が成功すると利用可能なデバイス数が **LONG** 値で返されます。

注意事項:

第 2 引数には、必ず [10][256] の文字配列を渡してください。
これより小さかった場合などに、戻り値に 0 が返ることがあります。

VB.NET 及び C#.NET ではこの関数を使用することは出来ません。
この関数が対応していない言語でデバイス名を取得したい場合は、
ArtCam_GetDeviceName を使用してください。

■ArtCam_SetCameraType

定義: **BOOL** ArtCam_SetCameraType(HACAM *hACam*, LONG *Flg*)

機能: 接続されている SATA カメラの機種をセットする

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Flg</i>	SATA カメラタイプコード

機能詳細:

Flg に指定したカメラタイプコードの SATA カメラを選択します。
Initialize 関数で ArtCamSdk Sata.dll を初期化した後に呼び出してください。

カメラタイプコードは以下のように定義されています。

機種名	カメラタイプコード
ARTSAT-0506LVDS	ARTCAM_CAMERATYPE_SATA_LVDS
ARTCAM-300MI-SATA	ARTCAM_CAMERATYPE_SATA_300MI
ARTCAM-500MI-SATA	ARTCAM_CAMERATYPE_SATA_500MI
ARTCAM-MV413-SATA	ARTCAM_CAMERATYPE_SATA_MV413
ARTCAM-800MI-SATA	ARTCAM_CAMERATYPE_SATA_800MI
ARTCAM-036MI-SATA	ARTCAM_CAMERATYPE_SATA_036MI
ARTCAM-267KY-SATA	ARTCAM_CAMERATYPE_SATA_150P

■ArtCam_GetCameraType

定義: **LONG** ArtCam_GetCameraType(HACAM *hACam*, LPBOOL *Error*)

機能: 接続されている SATA カメラの機種を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

接続されている SATA カメラのカメラタイプコードを取得します。

カメラのタイプコードは以下のように定義されています。

機種名	カメラタイプコード
ARTSAT-0506LVDS	ARTCAM_CAMERATYPE_SATA_LVDS
ARTCAM-300MI-SATA	ARTCAM_CAMERATYPE_SATA_300MI
ARTCAM-500MI-SATA	ARTCAM_CAMERATYPE_SATA_500MI
ARTCAM-MV413-SATA	ARTCAM_CAMERATYPE_SATA_MV413
ARTCAM-800MI-SATA	ARTCAM_CAMERATYPE_SATA_800MI
ARTCAM-036MI-SATA	ARTCAM_CAMERATYPE_SATA_036MI
ARTCAM-267KY-SATA	ARTCAM_CAMERATYPE_SATA_150P

■ArtCam_Width

定義: **LONG** ArtCam_Width(HACAM *hACam*)

機能: カメラ映像の幅を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

～DirectShow カメラの場合 (ArtCamSdk.dll 使用時)～

ArtCam_SetCaptureWindow で *Width* に指定した数値が **LONG** 値で返ります。

～キャプチャサイズが固定されたカメラの場合～

SDK 内部で設定されている基準サイズが **LONG** 値で返ります

(ArtCnvSdk.dll、ArtCamSdk_150P3.dll、ArtCamSdk_500P.dll 等)

～キャプチャサイズを変更できるカメラの場合～

ArtCam_SetCaptureWindowEx で *Heffective* に指定した数値が **LONG** 値で返ります。

■ArtCam_Height

定義: **LONG** ArtCam_Height(HACAM *hACam*)

機能: カメラ映像の高さを取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

～DirectShow カメラの場合 (ArtCamSdk.dll 使用時)～

ArtCam_SetCaptureWindow で *Height* に指定した数値が **LONG** 値で返ります。

～キャプチャサイズが固定されたカメラの場合～

SDK 内部で設定されている基準サイズが **LONG** 値で返ります

(ArtCnvSdk.dll、ArtCamSdk_150P3.dll、ArtCamSdk_500P.dll 等)

～キャプチャサイズを変更できるカメラの場合～

ArtCam_SetCaptureWindowEx で *Veffective* に指定した数値が **LONG** 値で返ります。

■ArtCam Fps

定義: **LONG** ArtCam_Fps(HACAM *hACam*)

機能: カメラのフレームレートを取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

設定されているフレームレートを **LONG** 値で取得します。
取得されるフレームレートは秒間のフレーム数を 10 倍した数値です。
秒間 30 フレームに指定している場合は 300 となります。
本関数は **ArtCamSdk.dll** (**DirectShow** カメラ) 専用です。

■ArtCam GetCameraInfo

定義: **BOOL** ArtCam_GetCameraInfo(HACAM *hACam*, LPCAMERAINFO *pInfo*)

機能: カメラの情報を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPCAMERAINFO	<i>pInfo</i>	カメラ情報

機能詳細:

接続されているカメラの設定可能値などの情報を取得します。
CAMERAINFO 型構造体は以下のように定義されています。

```
CAMERAINFO {
    LONG   lSize;           // 構造体のサイズ
    LONG   lWidth;          // カメラの有効最大幅
    LONG   lHeight;         // カメラの有効最大高さ
    LONG   lGlobalGainMin;  // グローバルゲインの最低値(使用不可能のカメラは-1)
    LONG   lGlobalGainMax;  // グローバルゲインの最低値(使用不可能のカメラは-1)
    LONG   lColorGainMin;   // カラーゲインの最低値(使用不可能のカメラは-1)
    LONG   lColorGainMax;   // カラーゲインの最低値(使用不可能のカメラは-1)
    LONG   lExposureMin;    // 露光時間の最低値(使用不可能のカメラは-1)
    LONG   lExposureMax;    // 露光時間の最低値(使用不可能のカメラは-1)
} *LPCAMERAINFO;
```

■ArtCam_SetIOPort

定義: **BOOL** ArtCam_SetIOPort(

HACAM *hACam*, BYTE *byteData*, LONG *longData*, DWORD *Reserve*)

機能: IO にデータを書き込む

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
BYTE	<i>byteData</i>	IO に書き込むデータ(バイトデータ)
LONG	<i>longData</i>	使用しません。0 を指定してください
DWORD	<i>Reserve</i>	使用しません。0 を指定してください

機能詳細:

IO ポートにデータ(8 ビット分)を書き込みます。最新版(07/07/11 現在)のドライバでは、下位(LSB)3、4 ビット目が OUT0、OUT1 ポートになっています。
例えば、*byteData* に 0x0C を書き込んだ場合、OUT0、1 ポートともに Hi レベルになります。

関数が成功したときは TRUE 若しくは 1 が、
失敗したときには FALSE 若しくは 0 が返ります。

■ArtCam_GetIOPort

定義: **BOOL** ArtCam_GetIOPort(

HACAM *hACam*, LPBYTE *byteData*, LPLONG *longData*, DWORD *Reserve*)

機能: IO からデータを読み込む

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBYTE	<i>byteData</i>	IO から読み込むデータ(バイトデータ)
LPLONG	<i>longData</i>	使用しません。NULL を指定してください
DWORD	<i>Reserve</i>	使用しません。0 を指定してください

機能詳細:

IO ポートからデータ(8 ビット分)を読み込みます。最新版(07/07/11 現在)のドライバでは、下位(LSB)1、2 ビット目が IN0、IN1 ポートになっています。

byteData の内、ポートのインプットレベルが Hi のビットに 1 が入ります。
例えば、IN0 のレベルが Low、IN1 のレベルが Hi の場合、*byteData* に 0x02 が入ります。

関数が成功したときは TRUE 若しくは 1 が、
失敗したときには FALSE 若しくは 0 が返ります。

注意事項:

上記の 2 つの関数は I/O 対応カスタマイズ済みカメラでのみ有効です。
ポートは、デバイスドライバロード時(OS 起動時または USB 接続時)に
Low レベルで初期化されます。ドライバがロードされるまでの電圧は不定です。

■ArtCam_SetSubSample

定義: **BOOL** ArtCam_SetSubSample(HACAM *hACam*, LONG *SubSampleMode*)

機能: 間引き転送モードを設定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>SubSampleMode</i>	サブサンプルモード

機能詳細:

間引き転送モードを設定します。
SubSampleMode に設定した数値によって映像を間引いて転送します。

SUBSAMPLE_1 全てのデータ
SUBSAMPLE_2 縦横1／2のデータ
SUBSAMPLE_4 縦横1／4のデータ
SUBSAMPLE_8 縦横1／8のデータ

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

注意事項:

間引き転送をする際に ArtCam_CallBackPreview を使用する際は、設定した
画像サイズよりも少ないデータが転送されることに注意してください。
また、設定できる転送モードは機種によって異なります。
CCD カメラには、基本的にこの機能はありません。

■ArtCam_GetSubSample

定義: **LONG** ArtCam_GetSubSample(HACAM *hACam*)

機能: 現在の間引き転送モードを取得します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

関数が成功すると、現在の間引き転送モードが **LONG** 値で返されます。

SUBSAMPLE_1 全てのデータ
SUBSAMPLE_2 縦横1／2のデータ
SUBSAMPLE_4 縦横1／4のデータ
SUBSAMPLE_8 縦横1／8のデータ

関数が失敗した場合、**-1** が返ります。

■ArtCam_SetWaitTime

定義: **BOOL** ArtCam_SetWaitTime(HACAM *hACam*, LONG *WaitTime*)

機能: ウェイトタイムを指定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>WaitTime</i>	ウェイトタイム

機能詳細:

ArtCam_Preview、ArtCam_CallBackPreview で動画を取得する際のフレーム間のウェイトタイムをミリ秒単位で指定します。デフォルトは 10 になっています。

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

注意事項:

WaitTime に小さな数値を設定すると、取りこぼしが減り、フレームレートは上がりますが、CPU の使用率も上がりますので注意してください。通常は 5～20 程度を指定してください。
WaitTime に大きな数値を設定すると、フレームレートは下がります。

■ArtCam_GetWaitTime

定義: **LONG** ArtCam_GetWaitTime(HACAM *hACam*)

機能: ウェイトタイムを取得します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

関数が成功したときは、戻り値に現在のウェイトタイムが **LONG** 値で取得されます。
関数が失敗したときは、戻り値に-1 が返ります。

■ArtCam_SetMirrorV

定義: **BOOL** ArtCam_SetMirrorV(HACAM *hACam*, **BOOL** *Flg*)

機能: 上下ミラー反転機能を設定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
BOOL	<i>Flg</i>	反転フラグ

機能詳細:

転送される画像の上下ミラー反転を有効／無効にします。

Flg に真(TRUE) を設定すると上下ミラー反転が有効に、
Flg に偽(FALSE)を設定すると上下ミラー反転が無効になります。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

注意事項:

カメラの機種によって、デフォルトのフラグが異なりますので注意してください。

■ArtCam_GetMirrorV

定義: **BOOL** ArtCam_GetMirrorV(HACAM *hACam*)

機能: 上下ミラー反転機能の現在状況を取得します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

上下ミラー反転の転送を行っているか取得します。
戻り値には、上下ミラー反転が有効になっていたら **TRUE** が、
上下ミラー反転が無効になっていたら **FALSE** が返されます。

注意事項:

カメラの機種によって、デフォルトのフラグが異なりますので注意してください。

■ArtCam_SetMirrorH

定義: **BOOL** ArtCam_SetMirrorH(HACAM *hACam*, **BOOL** *Flg*)

機能: 左右ミラー反転機能を設定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
BOOL	<i>Flg</i>	反転フラグ

機能詳細:

転送される画像の左右ミラー反転を有効／無効にします。

Flg に真(TRUE) を設定すると左右ミラー反転が有効に、
Flg に偽(FALSE)を設定するとミラー反転が無効になります。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

注意事項:

カメラの機種によって、デフォルトのフラグが異なりますので注意してください。

■ArtCam_GetMirrorH

定義: **BOOL** ArtCam_GetMirrorH(HACAM *hACam*)

機能: 左右ミラー反転機能の現在状況を取得します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
-------	--------------	------------

機能詳細:

左右ミラー反転の転送を行っているか取得します。
戻り値には、左右ミラー反転が有効になっていたら **TRUE** が、
左右ミラー反転が無効になっていたら **FALSE** が返されます。

注意事項:

カメラの機種によって、デフォルトのフラグが異なりますので注意してください。

■ArtCam_SetHalfClock

定義: **BOOL** ArtCam_SetHalfClock(HACAM *hACam*, LONG *Value*)

機能: カメラの動作クロックを半分にする

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Value</i>	ハーフクロック動作フラグ

機能詳細:

カメラの動作クロックを通常の半分にする事が可能です。
クロックを半分にすると、単純にフレームレートも半分程度になります。
Value に 1 を指定すると、ハーフクロックが有効になります。0 で無効です。

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

注意事項:

本関数は、次のような場合に使用してください。

- ・ 高いフレームレートが必要ではない場合
- ・ 複数のカメラを同時に使用する場合
- ・ 低スペックの PC で使用する場合
- ・ USB の転送が間に合わず、画像が崩れる場合
- ・ 10 ビット転送モードを使用する場合

本関数では直接クロック切り替えは行いません。
本関数にて設定した後、映像取得の関数で初期化した際に初めて
クロックの切り替えが反映します。クロックの切り替えは機種によって数秒掛かる場合があります。

■ArtCam_GetHalfClock

定義: **LONG** ArtCam_GetHalfClock(HACAM *hACam*, LPBOOL *Error*)

機能: カメラの動作クロック状態を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

現在のカメラの動作クロックの状態を **LONG** 値で取得します。
ハーフクロックが有効なら 1 が、無効なら 0 が返ります。

本関数が成功したときは、*Error* に **TRUE** が、
本関数が失敗したときは、*Error* に **FALSE** が返ります。

■ArtCam_SetAutoIris

定義: **BOOL** ArtCam_SetAutoIris(HACAM *hACam*, LONG *Value*)

機能: オートアイリスの状態を設定する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Value</i>	オートアイリス動作フラグ

機能詳細:

オートアイリス(自動明るさ補正)の有効/無効を設定します。
Value に指定する値は、*Value*=0 のときオートアイリス無効、
Value=1 のときシャッタースピードによるオートアイリスが有効、
Value=2 のときグローバルゲインによるオートアイリスが有効になります。

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

注意事項:

機種によって使用できる環境などに違いがありますので注意してください。
また、Sharpness や Brightness などのフィルタが有効になっている場合に
正しく動作しない場合がございますので、フィルタとの併用はなるべく避けてください。

■ArtCam_GetAutoIris

定義: **LONG** ArtCam_GetAutoIris(HACAM *hACam*, LPBOOL *Error*)

機能: オートアイリスの状態を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

オートアイリス(自動明るさ補正)の状態を取得します。
現在のオートアイリス(自動明るさ補正)の設定を **LONG** 値で取得します。
オートアイリスが無効なら 0 が、
シャッタースピードによるオートアイリスが有効なら 1 が、
グローバルゲインによるオートアイリスが有効なら 2 が返ります。

本関数が成功したときは、*Error* に **TRUE** が、
本関数が失敗したときは、*Error* に **FALSE** が返ります。

■ArtCam_SetSamplingRate

定義: **BOOL** ArtCam_SetSamplingRate(HACAM *hACam*, LONG *Value*)

機能: フレームレートと取り込みサイズを設定する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Value</i>	取り込みモード

機能詳細:

アナログビデオ信号のフレームレートと取り込みサイズを設定します。

Value に 0~3 を設定し、下記の取り込みモードを設定します。(NTSC)

<i>Value</i> = 0	720×470	30 フレーム
<i>Value</i> = 1	720×480	15 フレーム
<i>Value</i> = 2	640×470	30 フレーム
<i>Value</i> = 3	640×480	15 フレーム

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

注意事項:

本関数は、NTSC-USB2.0 コンバータ ARTCNVII (ArtCnvSdk.dll) の専用関数です。
映像の引き伸ばしは行いませんので、指定したサイズの信号が入力されていない場合は、
四隅に黒や緑のラインが入る場合があります。

■ArtCam_GetSamplingRate

定義: **LONG** ArtCam_GetSamplingRate(HACAM *hACam*, LPBOOL *Error*)

機能: フレームレートと取り込みサイズを取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

現在のフレームレートと取り込みサイズ設定を取得します。
戻り値は 0~3 の **LONG** 値で取得されます。
取得される値は ArtCam_SetSamplingRate の *Value* で設定される値と同じです。

本関数が成功したときは、*Error* に **TRUE** が、
本関数が失敗したときは、*Error* に **FALSE** が返ります。

■ArtCam_GetVideoFormat

定義: **LONG** ArtCam_GetVideoFormat(HACAM *hACam*, LPBOOL *Error*)

機能: 接続されているカメラの映像信号のタイプを取得

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

接続されているカメラの映像信号のタイプを取得します。

戻り値は 0～3 の **LONG** 値で取得されます。

取得される値と映像信号のタイプの対応は以下通りになります。

- 0 NTSC
- 1 PAL
- 2 PALM
- 3 SECAM

本関数が成功したときは、*Error*に **TRUE** が、

本関数が失敗したときは、*Error*に **FALSE** が返ります。

■ArtCam_WriteSromID

定義: **BOOL** ArtCam_WriteSromID(HACAM *hACam*, LONG *Address*, LONG *Value*)

機能: サブコードを登録します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Address</i>	書き込むアドレス
LONG	<i>Value</i>	書き込むデータ

機能詳細:

カメラ内の 8Byte メモリ空間 (EEPROM) にデータを書き込みます。
複数台のカメラを識別する必要があるときなどに使用してください。

Address には 0～7 のアドレスを指定できます。
Value には 0～255 のデータを指定できます。
それ以上の数値は切り捨てられるのでご注意ください

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

■ArtCam_ReadSromID

定義: **LONG** ArtCam_ReadSromID(HACAM *hACam*, LONG *Address*, LPBOOL *Error*)

機能: サブコードを読み取ります

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Address</i>	読み込むアドレス
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

カメラ内の 8Byte メモリ空間 (EEPROM) からデータを読み込みます。
Address に 0～7 のアドレスを指定すると、そのアドレスに書かれている値が
LONG 値で取得されます。

ArtCam_WriteSromID でカメラに固有のIDを登録しておけば、
その数値と本関数で読み込んだ数値を照らし合わせることで、
複数台カメラを使用した際に別々に管理する事が可能となります。

本関数が成功したときは、*Error* に **TRUE** が、
本関数が失敗したときは、*Error* に **FALSE** が返ります。

■ArtCam_WriteRegister

定義: **BOOL** ArtCam_WriteRegister(HACAM *hACam*, BYTE *Address*, DWORD *Value*)

機能: センサレジスタへの書き込みを行ないます

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
BYTE	<i>Address</i>	書き込むアドレス
DWORD	<i>Value</i>	書き込むデータ

機能詳細:

カメラセンサのレジスタにデータを書き込みます。

関数が成功したときは **TRUE** 若しくは **1** が、

失敗したときには **FALSE** 若しくは **0** が返ります。

注意事項:

レジスタ設定の詳細については、通常では公開しておりません。

この関数は、一部のカスタマイズカメラなどで使用することを想定しております。

通常カメラでレジスタの値を変更してしまった場合、予期せぬ不具合が起こる可能性がありますので、通常カメラでこの関数は使用しないでください。

■ArtCam_ReadRegister

定義: **DWORD** ArtCam_ReadRegister(HACAM *hACam*, BYTE *Address*, LPBOOL *Error*)

機能: センサレジスタ値の読み込みを行ないます

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
BYTE	<i>Address</i>	読み込むアドレス
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

カメラセンサの、指定したアドレスのレジスタ値を読み込みます。

本関数が成功したときは、*Error* に **TRUE** が、

本関数が失敗したときは、*Error* に **FALSE** が返ります。

注意事項:

レジスタ設定の詳細については、通常では公開しておりません。

この関数は、一部のカスタマイズカメラなどで使用することを想定しております。

■ArtCam_SetFilterValue

定義: **BOOL** ArtCam_SetFilterValue(HACAM *hACam*, LONG *FilterType*, LONG *Value*)

機能: 画像フィルター情報を設定する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>FilterType</i>	設定するフィルターの種類
LONG	<i>Value</i>	設定する数値

機能詳細:

ArtCam_SetImageDlg 及び ArtCam_SetAnalogDlg で設定できる項目を直接設定できます。
FilterType に関しては次ページの表または各言語の定義ファイルを参照してください。

関数が成功したときは **TRUE** 若しくは **1** が、
失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_GetFilterValue

定義: **LONG** ArtCam_GetFilterValue(

HACAM *hACam*, LONG *FilterType*, LPBOOL *Error*)

機能: 画像フィルター情報を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>FilterType</i>	設定するフィルターの種類
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

ArtCam_SetImageDlg 及び ArtCam_SetAnalogDlg で設定できる項目を直接取得できます。
Error に **NULL** を指定した場合、エラー情報は取得できません。
FilterType に関しては次ページの表または各言語の定義ファイルを参照してください。

本関数が成功したときは、*Error* に **TRUE** が入り、
FilterType に設定されている数値が **LONG** 値で返ります。
本関数が失敗したときは、*Error* に **FALSE** が入ります。

■ArtCam_Set***

定義: **BOOL** ArtCam_Set***(**HACAM** *hACam*, **LONG** *Value*)

機能: 画像フィルタを個別に設定します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LONG	<i>Value</i>	設定する数値

機能詳細:

ArtCam_SetFilterValue のラッパー関数です。
ArtCam_SetFilterValue の第二引数が予め設定された状態で関数がコールされます。

例えば、GlobalGain を 30 に変えたいときは、

ArtCam_SetFilterValue(hACam, ARTCAM_FILTERTYPE_GLOBAL_GAIN, 30)
とする代わりに、
ArtCam_SetGlobalGain(hACam,30)
とすることが出来ます。

■ArtCam_Get***

定義: **LONG** ArtCam_Get***(**HACAM** *hACam*, **LPBOOL** *Error*)

機能: 画像フィルタ設定を個別に取得します

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

ArtCam_GetFilterValue のラッパー関数です。
ArtCam_GetFilterValue の第二引数が予め設定された状態で関数がコールされます。

■ArtCam_SetSyncV

定義: **BOOL** ArtCam_SetSyncV(HACAM *hACam*, **BOOL** *Flg*)

機能: VD の極性を設定する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
BOOL	<i>Flg</i>	VD の極性

機能詳細:

LVDS コンバータの専用関数です。

Flg に **TRUE** を設定すると、入力信号の VD が **High** である期間に *Flg* に **FALSE** を設定すると、入力信号の VD が **Low** である期間に画像データがサンプリングされます。

実際には画像データのサンプリングは HD との AND 演算の結果でラッチされますので、次ページの ArtCam_SetSyncH 関数と組み合わせてお使いください。

関数が成功したときは **TRUE** 若しくは 1 が、
失敗したときには **FALSE** 若しくは 0 が返ります。

■ArtCam_GetSyncV

定義: **BOOL** ArtCam_GetSyncV(HACAM *hACam*, LP**BOOL** *Error*)

機能: 設定されている VD の極性を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LP BOOL	<i>Error</i>	エラー情報

機能詳細:

LVDS コンバータの専用関数です。

戻り値が **TRUE** の場合、入力信号の VD が **High** である期間に
戻り値が **FALSE** の場合、入力信号の VD が **Low** である期間に
画像データがサンプリングされます。

Error に **NULL** を指定した場合、エラー情報は取得できません。
FilterType に関しては次々ページの表または各言語の定義ファイルを参照してください。

本関数が成功したときは、*Error* に **TRUE** が入り、
極性が **BOOL** 値で返ります。
本関数が失敗したときは、*Error* に **FALSE** が入ります。

■ArtCam_SetSyncH

定義: **BOOL** ArtCam_SetSyncV(HACAM *hACam*, **BOOL** *Flg*)

機能: HD の極性を設定する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
BOOL	<i>Flg</i>	HD の極性

機能詳細:

LVDS コンバータの専用関数です。

Flg に **TRUE** を設定すると、入力信号の HD が **High** である期間に

Flg に **FALSE** を設定すると、入力信号の HD が **Low** である期間に

画像データがサンプリングされます。

実際には画像データのサンプリングは VD との AND 演算の結果でラッチされますので、前ページの ArtCam_SetSyncV 関数と組み合わせてお使いください。

関数が成功したときは **TRUE** 若しくは **1** が、

失敗したときには **FALSE** 若しくは **0** が返ります。

■ArtCam_GetSyncH

定義: **BOOL** ArtCam_GetSyncH(HACAM *hACam*, **LPBOOL** *Error*)

機能: 設定されている HD の極性を取得する

引数:

HACAM	<i>hACam</i>	カメラ識別用ハンドル
LPBOOL	<i>Error</i>	エラー情報

機能詳細:

LVDS コンバータの専用関数です。

戻り値が **TRUE** の場合、入力信号の HD が **High** である期間に

戻り値が **FALSE** の場合、入力信号の HD が **Low** である期間に

画像データがサンプリングされます。

Error に **NULL** を指定した場合、エラー情報は取得できません。

FilterType に関しては次次ページの表または各言語の定義ファイルを参照してください。

本関数が成功したときは、*Error* に **TRUE** が入り、

極性が **BOOL** 値で返ります。

本関数が失敗したときは、*Error* に **FALSE** が入ります。

■ 画像フィルター設定可能値（全カメラ共通）

※但し、ARTCNV II、ARTUST シリーズは除く

ARTCAM_FILTERTYPE_BRIGHTNESS						
明度を設定します	下限値:	-255	上限値:	255	規定値:	0
セッター: ArtCam_SetBrightness	ゲッター: ArtCam_GetBrightness					

ARTCAM_FILTERTYPE_CONTRAST					
コントラストを設定します		下限値:	-127	上限値: 127	規定値: 0
セッター:	ArtCam_SetContrast	ゲッター:	ArtCam_GetContrast		

ARTCAM_FILTERTYPE_HUE						
色相を設定します	下限値:	-360	上限値:	360	規定値:	0
セッター:	ArtCam_SetHue		ゲッター:	ArtCam_GetHue		

ARTCAM_FILTERTYPE_SATURATION						
彩度を設定します	下限値:	-255	上限値:	255	規定値:	0
セッター:	ArtCam_SetSaturation		ゲッター:	ArtCam_GetSaturation		

ARTCAM_FILTERTYPE_SHARPNESS			
シャープネスを設定します	下限値:	0	上限値: 30 規定値: 0
セッター: ArtCam_SetSharpness	ゲッター:	ArtCam_GetSharpness	
※このフィルタを有効にすると、CPU への負荷が上がり、フレームレートが低下する場合があります。			

ARTCAM_FILTERTYPE_BAYER_GAIN_R / BAYER_GAIN_G / BAYER_GAIN_B			
赤・緑・青ベイヤー値を設定します	下限値:	0	上限値:400(200) 規定値: 100
セッター: ArtCam_SetBayerGainRed	ゲッター:	ArtCam_GetBayerGainRed	
セッター: ArtCam_SetBayerGainGreen	ゲッター:	ArtCam_GetBayerGainGreen	
セッター: ArtCam_SetBayerGainBlue	ゲッター:	ArtCam_GetBayerGainBlue	
※ ベイヤー値は、ソフト的に色バランスを変更する機能です。通常、緑ベイヤー値は設定できません。 オートホワイトバランス(BAYER_GAIN_AUTO)を設定した場合、この値が変化します。 ※ 上限値は、機種によって 200 と 400 があります。201 以上設定する場合、PC の性能によって動作が不安定になる場合があります。その場合は 200 までの値を使用してください。			

ARTCAM_FILTERTYPE_BAYER_GAIN_AUTO			
オートホワイトバランスの ON/OFF を設定します		データ型: BOOL	規定値: FALSE
セッター: ArtCam_SetBayerGainAuto	ゲッター: ArtCam_GetBayerGainAuto		

ARTCAM_FILTERTYPE_BAYER_GAIN_RGB			
RGB ベイヤー値を一括で設定します。	下限値： 0	上限値： 0xffff	規定値：0x646464
セッター： ArtCam_SetBayerGainRGB	ゲッター： ArtCam_GetBayerGainRGB		
※R/G/B ベイヤー値に、それぞれ 0～255 の値を一括で設定します。通常は使用しません。			

ARTCAM_FILTERTYPE_GAMMA			
ガンマ値を設定します。(γ 1.0=100)	下限値:	0	上限値: 200 規定値: 100
セッター: ArtCam_SetGamma	ゲッター: ArtCam_GetGamma		
※このフィルタを有効にすると、CPU への負荷がかかり、フレームレートが低下する場合があります。			

ARTCAM_FILTERTYPE_GLOBAL_GAIN			
グローバルゲインを設定します	下限値: ****	上限値: ****	規定値: ****
セッター: ArtCam_SetGlobalGain	ゲッター: ArtCam_GetGlobalGain		
※設定できる値、デフォルト値はセンサにより異なります。			

ARTCAM_FILTERTYPE_COLOR_GAIN_R / GAIN_G1 / GAIN_G2 / GAIN_B	
ゲインを色ごとに設定します	設定範囲はグローバルゲインと同じです
セッター: ArtCam_SetColorGainRed	ゲッター: ArtCam_GetColorGainRed
セッター: ArtCam_SetColorGainGreen1	ゲッター: ArtCam_GetColorGainGreen1
セッター: ArtCam_SetColorGainGreen2	ゲッター: ArtCam_GetColorGainGreen2
セッター: ArtCam_SetColorGainBlue	ゲッター: ArtCam_GetColorGainBlue
※CMOS カメラ特有の機能です。通常はグローバルゲインの値に準じます。	

ARTCAM_FILTERTYPE_EXPOSURETIME			
シャッター速度を設定します	下限値: ****	上限値: ****	規定値: ****
セッター: ArtCam_SetExposureTime	ゲッター: ArtCam_GetExposureTime		
※設定できる値、デフォルト値はセンサにより異なります。			

ARTCAM_FILTERTYPE_BAYERMODE			
ベイヤーパターンを変更します	下限値:	0	上限値: 3 規定値: ****
セッター: ArtCam_SetBayerMode	ゲッター:	ArtCam_GetGlobalGain	
※正しい色が出力される規定値は、センサにより異なります。			

■グレースケールフィルター設定可能値 (DLLVer.1280 以降)

ARTCAM_FILTERTYPE_GRAY_MODE							
グレースケールのモードを設定します		下限値:	0	上限値:	2	規定値:	0
セッター: ArtCam_SetGrayMode		ゲッター: ArtCam_GetGrayMode					
0 = GRAY_NONE // 無効 ベイヤー配列のまま							
1 = GRAY_BAYERCONVERT // ベイヤー配列に GRAY GAIN と GRAY OFFSET で演算を加える							
2 = GRAY_GRAYSCALE // カラー変換した後、輝度情報のみにする							

ARTCAM_FILTERTYPE_GRAY_GAIN_R / GAIN_G1 / GAIN_G2 / GAIN_B			
色ごとのゲインをソフト的に調節します	下限値: 0	上限値: 400	規定値: 128
セッター: ArtCam_SetGrayGainRed	ゲッター: ArtCam_GetGrayGainRed		
セッター: ArtCam_SetGrayGainGreen1	ゲッター: ArtCam_GetGrayGainGreen1		
セッター: ArtCam_SetGrayGainGreen2	ゲッター: ArtCam_GetGrayGainGreen2		
セッター: ArtCam_SetGrayGainBlue	ゲッター: ArtCam_GetGrayGainBlue		

ARTCAM_FILTERTYPE_GRAY_OFFSET_R / OFFSET_G1 / OFFSET_G2 / OFFSET_B			
色ごとのオフセットをソフト的に調節します	下限値: -255	上限値: 255	規定値: 0
セッター: ArtCam_SetGrayOffsetRed	ゲッター: ArtCam_GetGrayOffsetRed		
セッター: ArtCam_SetGrayOffsetGreen1	ゲッター: ArtCam_GetGrayOffsetGreen1		
セッター: ArtCam_SetGrayOffsetGreen2	ゲッター: ArtCam_GetGrayOffsetGreen2		
セッター: ArtCam_SetGrayOffsetBlue	ゲッター: ArtCam_GetGrayOffsetBlue		

■画像フィルター設定可能値 (ARTCNVII)

ARTCAM_FILTERTYPE_BRIGHTNESS			
明度を設定します	下限値:	0	上限値: 255 規定値: 128
セッター: ArtCam_SetBrightness	ゲッター: ArtCam_GetBrightness		

ARTCAM_FILTERTYPE_CONTRAST			
コントラストを設定します	下限値:	0	上限値: 255 規定値: 128
セッター: ArtCam_SetContrast	ゲッター: ArtCam_GetContrast		

ARTCAM_FILTERTYPE_HUE			
色相を設定します	下限値:	0	上限値: 255 規定値: 0
セッター: ArtCam_SetHue	ゲッター: ArtCam_GetHue		

ARTCAM_FILTERTYPE_SATURATION			
彩度を設定します	下限値:	0	上限値: 255 規定値: 128
セッター: ArtCam_SetSaturation	ゲッター: ArtCam_GetSaturation		

ARTRAY Camera / Capture Module Software Developer Kit

Dynamic Link Library for Windows,2000,XP

株式会社 アートレイ

東京都杉並区高円寺北 1-17-5

上野ビル 5F

TEL 03-3389-5488

FAX 03-3389-5486

<http://www.artray.co.jp>

artray@artray.co.jp