

イベントベースカメラ API仕様書

2024/3/13

Grove Design Service

変更履歴

2024/3/13 初版

対応カメラ一覧

- CentyryArks 製 SilkeyEVCam(VGA)

概要

本 API は C/C++ から DLL にアクセスするためのアンセーフ関数インターフェイス、および C# からアクセスするためのセーフコード・クラスからなります。WPF 等の C# を用いたプロジェクトを使用する場合にはセーフコード・クラスを使用してください。

セーフコード、アンセーフコードの違いは理解しているものとします。

また、C# のクラスは複数のスレッドからアクセスした場合の排他制御を含みます。DLL 自体には複数スレッドからの排他制御は含まれていませんので、C/C++ から直接 DLL を呼び出す場合には、必要に応じて複数の CPU が同時にアクセスしないような仕組みをご用意ください。

USB に接続されたイベントベースカメラ・デバイスをオープンします。

```
bool EVC_UsbOpen(  
    HANDLE *FoundHandle,  
    PWINUSB_INTERFACE_HANDLE phWinUSBHandle  
);
```

[引数]

*FoundHandle	オープンした USB デバイスのデバイスハンドルを格納するポインタを指定します。
phWinUSBHandle	オープンした USB デバイスの WinUSB ハンドルを格納するポインタを指定します。

[戻り値]

USBデバイスを**Open**できた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

USBデバイスをクローズします。

```
bool EVC_UsbClose(  
    HANDLE FoundHandle,  
    WINUSB_INTERFACE_HANDLE hWinUSBHandle  
);
```

[引数]

FoundHandle	オープンしたときに取得した USB デバイスの デバイス ハンドルを指定します。
hWinUSBHandle	クローズする USB デバイスの WinUSB ハンドルを指定します。

[戻り値]

USBデバイスを**Close**できた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

カメラから取得したメーカー名、製品名を取得します。

```
bool EVC_GetDeviceInfo(  
    WINUSB_INTERFACE_HANDLE hWinUSBHandle,  
    UCHAR* pMakerStr,  
    UCHAR* pProductStr  
);
```

[引数]

hWinUSBHandle	オープンしたときに取得した WinUSB の ハンドルを指定します。
pMakerStr	メーカー名を格納するバッファへのポインタを指定します。
pProductStr	製品名を格納するバッファへのポインタを指定します。

[戻り値]

メーカー名、製品名を取得できた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

カメラが接続しているUSBの現在の速度（モード）を取得します。

```
bool EVC_GetSpeed(  
    WINUSB_INTERFACE_HANDLE hDeviceHandle,  
    UCHAR* pDeviceSpeed  
);
```

[引数]

hWinUSBHandle	オープンしたときに取得した WinUSB の ハンドルを指定します。
pDeviceSpeed	速度値を格納するバッファへのポインタを指定します。 1 : Low Speed (USB1.0) 2 : Full Speed (USB2.0) 3 : High Speed (USB 3.0)

[戻り値]

測度を取得できた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

カメラのシリアル番号を取得します。

```
bool EVC_GetSerialNumber(  
    WINUSB_INTERFACE_HANDLE hDeviceHandle,  
    CHAR* pDeviceSpeed  
);
```

[引数]

hWinUSBHandle オープンしたときに取得した **WinUSB** の ハンドルを指定します。
pDeviceSpeed シリアル番号文字列を格納するバッファへのポインタを指定します。
 8バイト以上のバッファを指定してください。

[戻り値]

シリアル番号を取得できた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

カメラに接続し、初期化を行います。

```
bool EVC_Connect(  
    WINUSB_INTERFACE_HANDLE hWinUSBHandle  
);
```

[引数]

hWinUSBHandle オープンしたときに取得した **WinUSB** の ハンドルを指定します。

[戻り値]

カメラに接続し、初期化できた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

[補足]

イベントデータに含まれる時刻は本関数で 0 に戻ります。**EVC_Stop()**
して**EVC_Start()**した場合には、前回止めた時刻を基準に進みます。
時刻を最初に戻すには再度**EVC_Connect()**する必要があります。

カメラの画像取り込みを開始（再開）します。

```
bool EVC_Start(  
    WINUSB_INTERFACE_HANDLE hWinUSBHandle  
);
```

[引数]

hWinUSBHandle オープンしたときに取得した **WinUSB** の ハンドルを指定します。

[戻り値]

カメラから画像の取り込みを開始できた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

[補足]

EVC_Stop()で停止（一時停止）した取り込みを再開することもできます。

カメラの画像取り込みを停止（一時停止）します。

```
bool EVC_Stop(  
    WINUSB_INTERFACE_HANDLE hWinUSBHandle  
);
```

[引数]

hWinUSBHandle オープンしたときに取得した **WinUSB** の ハンドルを指定します。

[戻り値]

カメラから画像の取り込みを停止できた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

カメラからイベントデータを読み出します。

```
bool EVC_ReadData(  
    WINUSB_INTERFACE_HANDLE hWinUSBHandle,  
    int* length,  
    UCHAR* pData  
);
```

[引数]

hWinUSBHandle	オープンしたときに取得した WinUSB の ハンドルを指定します。
Length	実際に取得したバイト数を返します。
pData	データを格納するバッファへのポインタを指定します。 バッファの大きさは 128KBytes が必要です。

[戻り値]

イベントデータが読み出しできた場合**TRUE**を返します。
それ以外は**FALSE**を返します。

USB に接続されたイベントベースカメラ・デバイスをオープンします。

```
public bool UsbOpen()
```

[戻り値]

USBデバイスをOpenできた場合trueを返します。
それ以外はfalseを返します。

USBデバイスをクローズします。

```
public void UsbClose()
```

カメラから取得したメーカー名、製品名を取得します。

```
public bool GetDeviceInfo(out string makerStr, out string productStr)
```

[引数]

makerStr	メーカー名を示す string 型の文字列を返します
productStr	製品名を示す string 型の文字列を返します

[戻り値]

メーカー名、製品名を取得できた場合trueを返します。
それ以外は false を返します。

カメラが接続しているUSBの現在の速度（モード）を取得します。

```
public bool GetSpeed(out int speed)
```

[引数]

speed	現在の USB 速度を示す int 型の値を返します EVC.LOW_SPEED : Low Speed (USB1.0) EVC.FULL_SPEED : Fill Speed (USB2.0) EVC.HIGH_SPEED : High Speed (USB 3.0)
-------	--

[戻り値]

現在のUSB速度を取得できた場合trueを返します。
それ以外は false を返します。

カメラのシリアル番号を取得します。

```
public bool GetSerialNumber(out string sSerialNumber)
```

[引数]

sSerialNumber カメラのシリアル番号を示す **string** 型の文字列を返します

[戻り値]

シリアル番号を取得できた場合**true**を返します。
それ以外は **false** を返します。

カメラに接続し、初期化を行います。

```
public bool Connect()
```

[戻り値]

初期化できた場合 **true** を返します。それ以外は **false** を返します。

[補足]

イベントデータに含まれる時刻は本関数で 0 に戻ります。**Stop()**
して**Start()**した場合には、前回止めた時刻を基準に進みます。
時刻を最初に戻すには再度**Connect()**する必要があります。

カメラの画像取り込みを開始（再開）します。

```
public bool Start()
```

[戻り値]

開始できた場合 **true** を返します。それ以外は **false** を返します。

[補足]

Stop()で停止（一時停止）した取り込みを再開することもできます。

カメラの画像取り込みを停止（一時停止）します。

```
public bool Stop()
```

[戻り値]

停止できた場合 **true** を返します。それ以外は **false** を返します。

カメラからイベントデータを読み出します。

```
public bool ReadData(out byte[] pData)
```

[引数]

pData 読み出したイベントデータを返します。配列は読み出した長さを持つ。

[戻り値]

停止できた場合 **true** を返します。それ以外は **false** を返します。