NetVision

UVC SDK Manual

(Windows)

V2.50

2023.02.20

NetVision Corp.

目次

改版履歴

[2016/06/20] V1.00 新規作成

[2017/07/10] V2.00 標準版ソフトのバージョンアップを反映

[2018/03/09] V2.10 Raw8 対応、NVShowFrameNumDLL の追加

[2021/12/14] V2.20 I2C 送受信データサイズに関する説明文の誤りを修正 (最大 30 バイト -> 29 バイト) 、SVM-MIPI を SVM-06 に変更、「NVCap」、「NVRawDLL」、「SVMCtl_I2C」を最新のバージョンに更新

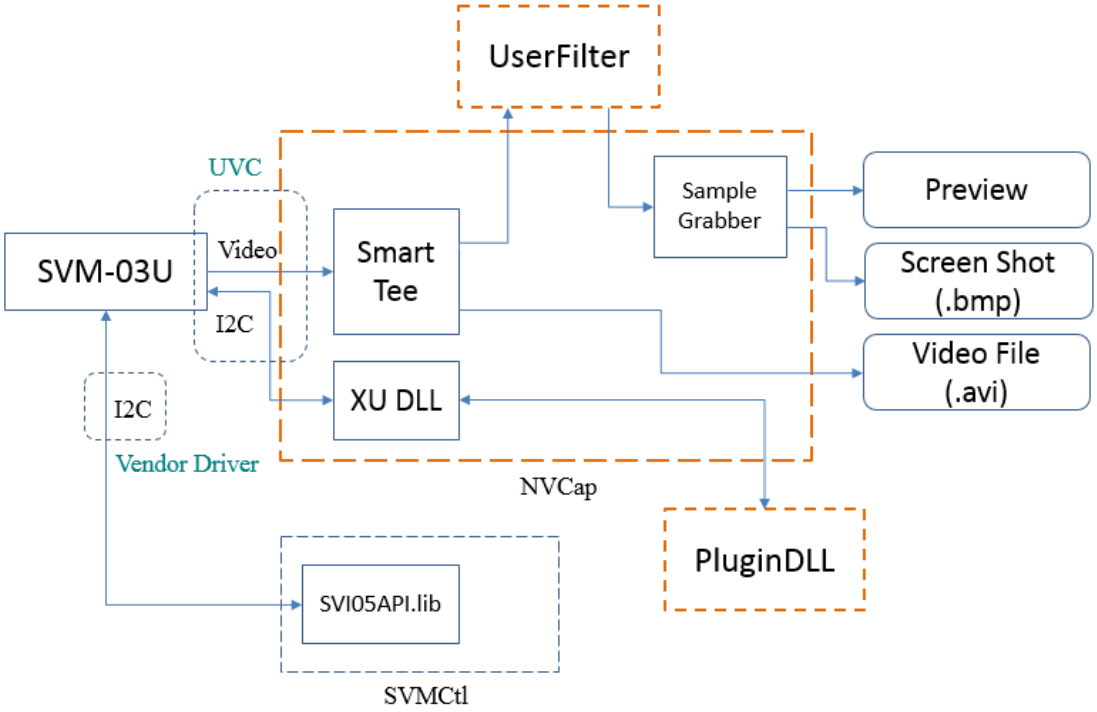
[2023/02/02] V2.50 各プロジェクトを最新のソースコードに更新、社内バージョン番号と本ドキュメントのバージョン番号を連動

1. 概要

NetVision UVC SDK は、NetVision SVM-03U ボードや SVM-06 ボード等、弊社 USB Video Class (UVC) 準拠 USB3.0 キャプチャボード（以下 SVM ボードと表記）を使用するためのソフトウェア環境一式を含む SDK です。

本 SDK は Windows 版 SDK と Linux 版 SDK によって構成されています。Windows 版 SDK は Microsoft の DirectShow を用いたキャプチャソフト、ターゲットデバイスとの間で I2C 通信を行うためのライブラリ、キャプチャソフトで映像処理や I2C 通信を行うためのサンプルプラグインを含みます。Linux 版 SDK は OpenCV ライブラリを用いたキャプチャソフト、再生ソフト、I2C 通信を行うためのライブラリ、ボード設定用のユーティリティソフトを含みます。

ライブラリやサンプルはすべて C++ 言語で書かれています。また、Windows 版 SDK のビルドには MFC に対応した Microsoft Visual Studio 2008 以降が必要です。Linux 版 SDK は gcc およびオープンソースのライブラリを使用してビルドできるようになっています。



Windows 版 SDK を使用した SVM ボードのキャプチャ環境の構成図を上図に示します。UVC を使用したキャプチャは図中に橙色で示した「NVCap」「UserFilter」および「PluginDLL」の 3 種類のアプリケーションおよび DLL によって動作し、映像信号の受信と I2C 通信が可能です。また、UVC 準拠の機能とは独立に、弊社独自のドライバを使用したユーティリティソフト「SVMCtl」を使用した I2C 通信も対応しています。

「NVCap」は Windows 版 SDK の核となる SVM ボード用キャプチャソフトで、4 台までの SVM ボードの同時使用に対応しています。NVCap の基本機能はリアルタイムのプレビュー、非圧縮 .avi 形式でのキャプチャおよび .bmp 形式でのスクリーンショットです。このほか、SVM ボードの I2C バスをコントロールして、ターゲットデバイスとの間の I2C 通信もサポートしています。

これらの機能は UVC 準拠で動作するため、Windows の場合 DirectShow と OS 標準のドライバにより動作します。SDK 付属の NVCap だけでなく、UVC カメラに対応したサードパーティー製キャプチャソフトを使用することもできます。I2C のコントロールは UVC の Extension Unit 経由で行われる仕組みであり、これには NVCap とは別に SVM ボード用の Extension Unit ライブラリ（図中「XU DLL」で示した「Extension Unit DLL」）のインストールが必要になります。「Extension Unit DLL」は本 SDK に付属しています。SVM ボードの Extension Unit を複雑な DirectShow の知識なしに扱うために、本 SDK には Extension Unit のラッパークラスを用意しており、NVCap 以外のキャプチャソフトにも SVM ボードの I2C 通信機能などが容易に組み込めるようになっています。

さらに、NVCap のプラグイン機能として「UserFilter」や「PluginDLL」があります。「UserFilter」は入力された映像データに対して処理を加えるためのプラグインです。「PluginDLL」は I2C 送受信などデバイスとの間で映像以外の通信を行うためのダイアログを表示するプラグインです。両プラグインとも DLL として動作するため、NVCap とは別にユーザ独自の「UserFilter」や「PluginDLL」を開発することにより、キャプチャされた映像への画像処理やターゲットデバイスや SVM ボードの設定を行うシステムを作成することができます。

「SVMCtl」は SVM ボード標準付属のユーティリティソフトで、I2C 通信の他に ボード設定や FW アップデートなどの機能があります。本 SDK には、SVMCtl の I2C 部分のみを取り出したプロジェクト「SVMCtl_I2C」を付属しています。

1.1. プロジェクト構成

Windows 版 SDK は、

- NVCap (UVC キャプチャソフト)
- Extension Unit DLL
- PluginDLLTemplate (PluginDLL テンプレートプロジェクト)
- PluginDLL_IMX219 (IMX219 = Raspberry Pi Camera 用 PluginDLL サンプルプロジェクト
- NVFlipDLL (UserFilter サンプルプロジェクト)
- NVRawDLL (UserFilter サンプルプロジェクト)
- NVShowFrameNumDLL (UserFilter サンプルプロジェクト)
- SVMCtl_I2C

によって構成されています。それぞれのプロジェクトの詳細は 2 章以降に説明します。

1.2. 推奨動作環境

名前	値	備考
OS (Windows 版 SDK)	Windows 7 / 8.1	
OS (Linux 版 SDK)	Ubuntu 16.04 LTS	
CPU	Intel Core i5 以降	
Memory	3GB 以上	

主記憶装置	SSD 使用	録画機能を使用する場合
シリアルインタフェース	USB 3.0 ポート	マザーボード搭載のコントローラを推奨 PCI カードや USB ハブは相性問題あり
コンパイラ (Windows)	Microsoft Visual C++ 2008 Professional	要 MFC サポート、Microsoft DirectShow SDK、Windows DDK
コンパイラ (Linux)	gcc	詳細は Linux 版マニュアル参照

- USB ハブを介した多チャンネル取り込みの場合 TI 社製コントローラを使用されることを推奨します。

本 SDK のビルドには

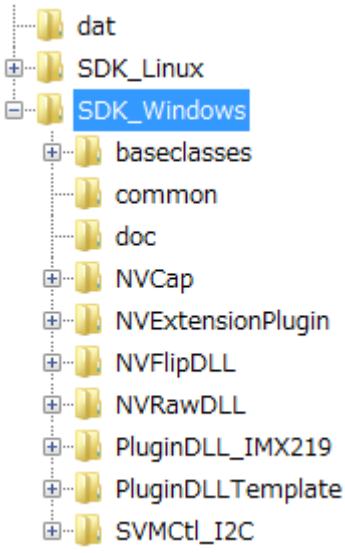
「Microsoft Visual C++ 2008 Professional Edition」

「Windows Driver Kit (WDK) 8.1」

「DirectX Software Development Kit (DirectX SDK)」

のインストールが必要です。また、環境に応じて各プロジェクトのパスの再設定が必要になる場合があります。

2. フォルダ構成



フォルダ名	説明

dat	センサ設定ファイルを格納しています。
Linux_SDK	Linux 用 SDK を格納しています。
Windows_SDK	Windows 用 SDK を格納しています。
+ baseclasses	各プロジェクトから参照する Direct Show 基底クラスです。
+ common	baseclasses のプロジェクトが参照するコードを格納します。
+ doc	ドキュメント類を格納しています。
+ NVCap	NVCap プロジェクト一式です。
+ NVExtensionPlugin	NVExtensionPlugin プロジェクト一式です。
+ NVFlipDLL	NVFlipDLL プロジェクト一式です。
+ NVRawDLL	NVRawDLL プロジェクト一式です。
+ NVShowFrameNumDLL	NVShowFrameNumDLL プロジェクト一式です。
+ PluginDLL_IMX219	IMX219 用の PluginDLL プロジェクト一式です。
+ PluginDLLTemplate	PluginDLL のテンプレートプロジェクトです。
+ SVMCtl_I2C	SVMCtl の I2C 部分のみのプロジェクト一式です。

各プロジェクトが別のプロジェクトのフォルダ内のファイルを参照していることがあるので、フォルダ構成を変更せず、SDK_Windows フォルダごとローカルにコピーして使用してください。

2.1. センサ設定ファイル

場所: ./dat/

2.1.1. 内容

SVMCtl (Windows)、NVCap (Windows) アプリケーションや SVMSender (Linux) はセンサ設定ファイルに書かれた I2C シーケンスをセンサに送信することができます。dat フォルダにはこのセンサ設定ファイルのサンプルをいくつか格納しています。

ファイル名	説明
imx219_720p.txt	Raspberry Pi Camera v2 を 1280x720 / 47.5 fps / Raw10 / MIPI 2 Lane で初期化するための設定ファイルです。
imx219_1080p.txt	Raspberry Pi Camera v2 を 1920x1080 / 47.5 fps / Raw10 / MIPI 2 Lane で初期化するための設定ファイルです。
ov5647_1080p.txt	Raspberry Pi Camera を 1920x1080 / 30 fps / Raw10 / MIPI 2 Lane で初期化するための設定ファイルです。
ov5642_720p_30fps_vs_neg_uyvy.txt	OV5642 (Omnivision) を 1280x720 / 30 fps / UYVY / 8bit で初期化するための設定ファイルです。

ov5642_vga_30fps_vs_neg_uyvy.txt	OV5642 (Omnivision) を 640x480 / 30fps / UYVY / 8bit で初期化するための設定ファイルです。

2.2. 関連ドキュメント

場所: SDK_Windows/doc/

ファイル名	説明
NV_UVC_SDK_Manual(Windows).pdf	本書です。
NVCap ソフトウェアマニュアル-V151.pdf	キャプチャソフト「NVCap」のソフトウェアマニュアルです。
SVMCtl ソフトウェアマニュアル_9.5.pdf	SVM ボード用設定ユーティリティ「SVMCtl」のソフトウェアマニュアルです。
NVFilter_Instruction.pdf	UserFilter に関する説明書です。
Extension_Unit_Specification_v2.0_Release.pdf	SVM ボードの Extension Unit の仕様書です（I2C 関連部分のみ抜粋）。

3. Windows 版 SDK

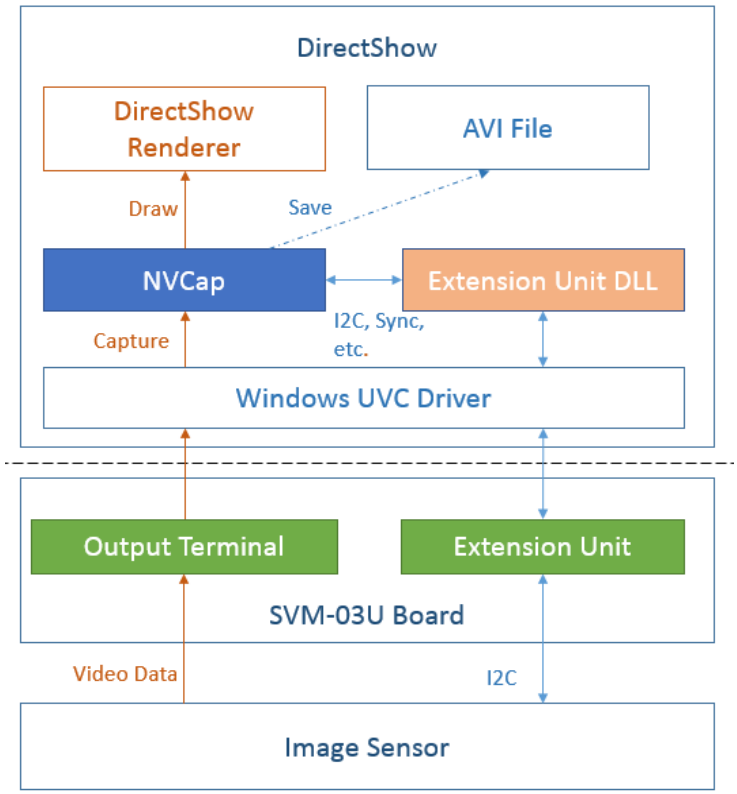
場所: /SDK_Windows/

3.1. UVC キャプチャソフト「NVCap」

場所: SDK_Windows/NVCap

3.1.1. 概要

NVCap は Microsoft DirectShow を用いた SVM-03U / SVM-06 等ボード用のマルチチャンネルキャプチャソフトで、最大 4ch までのボードから同時にプレビュー、非圧縮 .avi 形式での録画、bmp 形式での静止画の保存、I2C 通信などの操作を行うことができます。



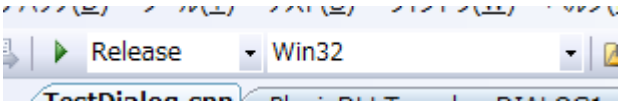
ソフトウェア構成図を上図に示します。NVCap にはビデオデータのキャプチャ機能、保存機能、I2C 送受信機能等がありますが、キャプチャ・保存に関しては OS 標準ドライバのみで動作可能です。センサへの I2C 送受信等は Extension Unit を通して実行されます。PC にインストールされた「Extension Unit DLL」を通して Windows 標準ドライバを経由することで、DirectShow の枠組みの中で行っています。Extension Unit DLL はあらかじめレジストリに登録することで、DirectShow から自動的に呼び出されるようになっています。Extension Unit DLL のインストール方法については後述します。

NVCap の使用方法については、「NVCap ソフトウェアマニュアル」も参照してください。

3.1.2. ビルド方法

NVCap や他のプロジェクトは、baseclasses フォルダ内のプロジェクトからビルドされるライブラリ（strmbase.lib または strmbasd.lib）を参照します。ビルド前に、お使いの環境で baseclass プロジェクトをリビルドして、ライブラリ strmbase.lib が出力されたことを確認してください。

Visual Studio 2008 よりプロジェクトを開き、プロジェクトの構成を Release-Win32 もしくは Release - x64 に設定してビルドを行います。



baseclasses プロジェクトや NVCap プロジェクトがビルドできない場合、Microsoft DirectShow SDK もしくは Windows DDK が正常にインストールされていない可能性があります。

ビルドに成功すると、Release-Win32 の場合 Release フォルダ内に NVCap.exe が生成されます。Release-x64 の場合は、x64/Release フォルダ内に NVCapx64.exe が生成されます。フォルダ内の .ax ファイルは Raw 形式や RGB 形式のキャプチャ時に NVCap 内部で使用される Direct Show フィルタなので、削除しないでください。

NVCap とは別にユーザが拡張できるプラグインとして、NVCap フォルダ内の Filter フォルダには「UserFilter」、PluginDLL フォルダには 「PluginDLL」を格納します。これらは DLL なので、32bit / 64bit それぞれ別々にビルドされた DLL ファイルが必要です。UserFilter、PluginDLL の説明は前述した通り、本 SDK にはいくつかのサンプルプロジェクトを格納しています。

3.2. Extension Unit DLL

場所: SDK_Windows/NVExtensionPlugin/

主要なファイル:

ファイル名	説明
Release/ NVExtensionPlugin.dll	32 bit 版 Extension Unit DLL の本体です。 32 bit OS の場合こちらを使用します。 NVCap 等から Extension Unit 機能を使用する場合、DirectShow がこの DLL を呼び出すためにレジストリへの登録が必要です。
x64/Release/ NVExtensionPlugin_x64.dll	64 bit 版 Extension Unit DLL の本体です。 64 bit OS の場合、こちらを使用します。
Release/ | x64/Release/ ExtensionDLL_Install.bat	DLL レジストリ登録用バッチファイルです。 「管理者として実行」すると、Extension Unit DLL がシステムフォルダにコピーされたうえでレジストリに登録されます。
NVExtensionPlugin/ NVExtensionPlugin_i.h	DLL のインタフェース用のヘッダファイルです。 Extension Unit 機能を使用するためには、本ファイルのインクルードが必要です。

NVExtensionPlugin/ NVExtensionPlugin_i.c	DLL で使用する GUID 等を定義したソースファイルです。 キャプチャソフト等から直接本 DLL を使用する場合に、このファイルを使用します。

3.2.1. 概要

Windows 環境において DirectShow が Extension Unit 機能を利用するには、ベンダ側で作成した Extension Unit Plugin DLL を DirctShow が呼び出します。本 SDK 付属の「Extenision Unit DLL」は、SVM ボード用に作成された Extension Unit Plugin DLL です。Extension Unit Plugin DLL の詳細説明は、以下に挙げた Microsoft のドキュメントを参照してください。

https://msdn.microsoft.com/en-us/library/windows/hardware/ff568656(v=vs.85).aspx

通常、標準版 SVM ボードのアプリケーション CD には SVM ボード用にビルド済みの DLL を同梱しています。SDK にはこの DLL 本体の他に、DLL を任意のプロジェクトで使用するためのヘッダファイル、ソースファイルを含むプロジェクト一式を格納しています。

なお Extension Unit Plugin DLL を利用するためには、regsvr32 ツールであらかじめ DLL を PC にインストールする必要があります。DLL と同じフォルダにある「ExtensionDLL_Install.bat」を「管理者として実行」するか、管理者権限で起動したコマンドプロンプトより、DLL と同じフォルダにカレントディレクトリを移動した上で

```
regsvr32 NVExtensionPlugin.dll
```

と入力することで、インストール（レジストリへの登録）が完了します。アンインストール（レジストリからの登録解除）はコマンドプロンプトより

```
regsvr32 /u NVExtensionPlugin.dll
```

と入力してください。なお、x64 版では DLL のファイル名は「NVExtensionPlugin_x64.dll」となります。

3.3. Extension Unit ラッパークラス

場所: SDK_Windows/NVCap/ExtensionUnitManager.h

3.3.1. 概要

NVCap やその他のキャプチャソフトから SVM ボードやイメージセンサとの I2C 通信に必要な UVC の Extension Unit を簡単に使用するために、Windows 版 SDK では Extension Unit ラッパークラス（CNVExtensionManager クラス）を用意しています。NVCap

や SDK 付属の Plugin DLL プロジェクトでは、CNVExtensionManager クラスを使用して I2C 通信や SVM ボードのステータス受信をしています。

Extension Unit ラッパークラスは NVCap のプロジェクトに含まれており、ヘッダファイル ExtensionUnitManager.h に関数の記述がされています。このクラスを使用するプロジェクトから ExtensionUnitManager.h をインクルードします。

他のキャプチャソフトから使用する場合、クラスのインスタンスを生成する際、コンストラクタの引数に IBaseFilter へのポインタで定義される、あらかじめインスタンス化されたキャプチャデバイス（SVM ボード）へのインタフェースを指定してください。PluginDLL より使用する場合は、NVCap から PluginDLL の ShowDialog 関数を呼び出す際に引数としてキャプチャデバイスへのインタフェースが渡されるので、このポインタを使用して CNVExtensionManager クラスをインスタンス化してください。詳しい利用方法は PluginDLL のソースコードを参照してください。

3.3.2. 関数一覧

通常使用する関数を下記に示します。

関数名	説明
CNVExtensionManager	コンストラクタです。IBaseFilter として定義されるキャプチャデバイス (SVM ボード) へのインタフェースへのポインタを引数に指定します。
~CNVExtensionManager	デストラクタです。
SendI2CData	I2C データを送信します。 引数 - int addr 7bit の I2C デバイスアドレスを指定します。最上位 bit は 0 を指定してください。 - const unsigned char* data 送信するデータの配列を指定します。 1 回に送信できるデータ長は最大 29 バイトとなります。 - int length 引数 data のデータ長 [byte] を指定します。 返り値 0: 成功 それ以外: 失敗 備考

	SVM ボードの FW バージョン 71 以前では、ターゲットデバイスから NACK を受信しても本関数は成功を返します。
ReceiveI2C	I2C データを受信します。 引数 - int addr 7bit の I2C デバイスアドレスを指定します。最上位 bit は 0 を指定してください。 - unsigned char* data 受信されたデータが格納される配列をしてします。 1 回に受信できるデータ長は最大 29 バイトとなります。 - int length 引数 data のデータ長 [byte] を指定します。 - const unsigned char* preamble 受信レジスタアドレスなど、受信コマンドの前に送信するデータ (preamble) を指定します。 - int preambleSize preamble のデータ長 [byte] を指定します。 - int receiveWait = I2C_WAIT_1 コマンド送出から受信データをボードから読み込むまでの待ち時間を指定します。 返り値 0: 成功 それ以外: 失敗 備考 引数 preamble は後述する「I2C パケットフォーマット」図中の Register Address に相当します。 SVM ボードの FW バージョン 71 以前では、ターゲットデバイスから NACK を受信しても本関数は成功を返します。
GetFPGARegister	SVM ボードの FPGA のレジスタ値を取得します。 FPGA レジスタマップは通常公開していませんが、カスタム対応等で一部分のみユーザに開示することがあります。 引数 - unsigned int addr 32bit の FPGA レジスタアドレスを設定します。

	- unsigned int* value レジスタ値を格納する変数へのポインタを指定します。 返り値 0: 成功 それ以外: 失敗
SetFPGARegister	SVM ボードの FPGA のレジスタ値を設定します。 引数 - unsigned int addr 32bit の FPGA レジスタアドレスを設定します。 - unsigned int value 新たに設定するレジスタ値を指定します。 返り値 0: 成功 それ以外: 失敗

3.3.3. I2C パケットフォーマット

S: Start Bit

P: Stop Bit

A: ACK

N: NACK

3.3.4. Extension Unit 詳細仕様

SVM ボードで実装している Extension Unit の実装仕様を知りたい場合は、付属のドキュメント「Extension_Unit_Specification_v2.0_Release.pdf」を参照してください。

この SDK は ユーザが Extension Unit についての知識なしで Extension Unit の機能を使用できるように実装されています。

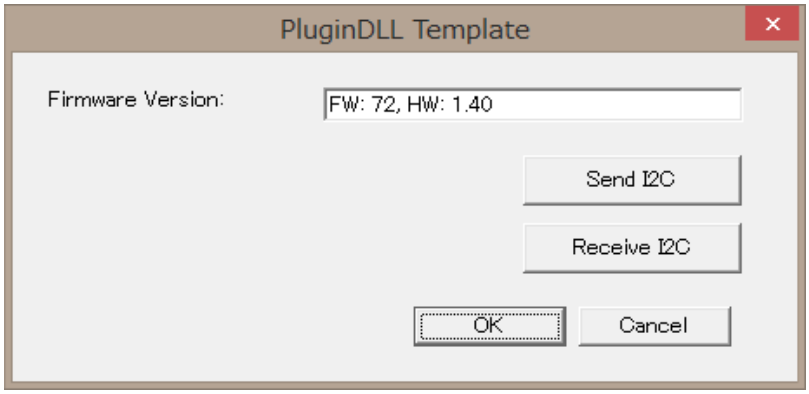
3.4. PluginDLL テンプレートプロジェクト

場所: SDK_Windows/PluginDLLTemplate/

主要なファイル:

ファイル名	説明
PluginDLLTemplate.cpp PluginDLLTemplate.h	ダイアログ画面の実装です。
TestDialog.cpp TestDialog.h	DLL のエクスポート関数の実装です。
PluginDLLTemplate.def	DLL のエクスポート定義ファイルです。

3.4.1. 概要

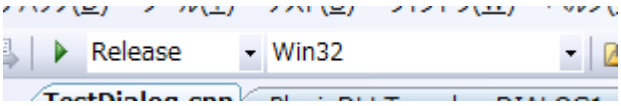


PluginDLL はキャプチャソフト「NVCap」から呼び出して使用される、I2C や SVM ボードと通信するための DLL です。NVCap とは独立にこの DLL を開発することで、イメージセンサへの設定送受信のための画面を実装することができます。

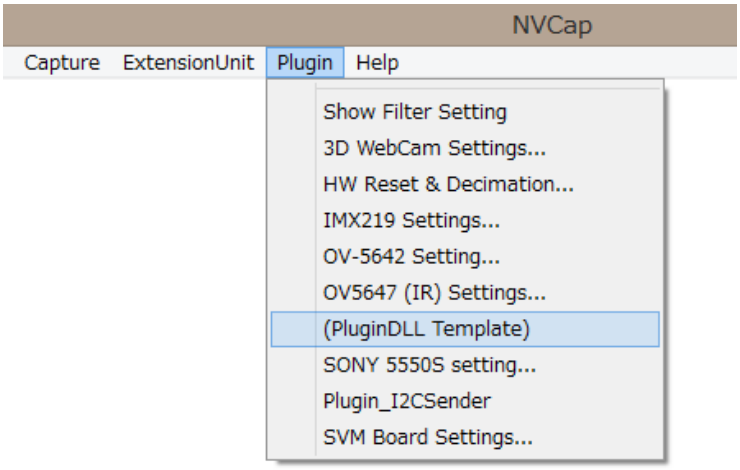
この PluginDLL テンプレートプロジェクトは、前述の Extension Unit ラッパークラスを用いた簡単なダイアログ(上図参照)を表示する DLL をビルドします。ユーザはこのプロジェクトを元に、使用するセンサや目的に応じたダイアログを開発することができます。

3.4.2. ビルド方法

Visual Studio 2008 よりプロジェクトを開き、プロジェクトの構成を Release-Win32 もしくは Release - x64 に設定してビルドを行います。



Release-Win32 でビルドした場合「Release/PluginTemplate.dll」 Release-x64 でビルドした場合「x64/Release/PluginTemplate.dll」が生成されます。この DLL ファイルを NVCap.exe が置かれたフォルダ内の 「PluginDLL」フォルダにコピーすることで、NVCap から プラグイン DLL を呼び出すことができます。x64 と x86 (Win32) のファイルを混同させると正常に動作しません。「PluginDLL」フォルダ内の DLL ファイルは NVCap 起動時にスキャンされ、NVCap の「Plugin」メニュー内に項目を生成します。



(複数の PluginDLL を入れた例)

3.4.3. 機能説明

- Firmware Version

接続されている SVM ボードの FW バージョンと FPGA バージョンを問い合わせて表示します。正常に通信が行われなかった場合、不正な数字が表示されます。

- Send I2C

接続されている SVM ボードに対し I2C データ送信要求を送信します。送信するデータやデバイスアドレスは、TestDialog.cpp 内の OnBnClickedSend() 関数内に定義されています。

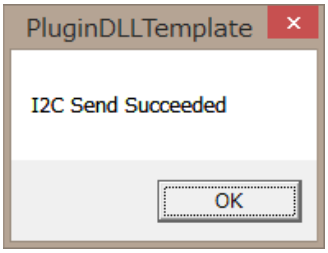
```
void CTestDialog::OnBnClickedSend()
{
    // I2C データの送信
    if(m_pBaseFilter){
        CNVExtensionManager extManager(m_pBaseFilter);
        if(extManager.IsOpen() == 0){
            MessageBox(_T("Can't Get ExtensionUnit Interface."), _T("Error"), MB_OK);
            return;
        }

        unsigned char sendAddr = 0x3D; // Device addr
        unsigned char sendData[3] = {0x20, 0x00, 0x00}; // data

        int ret = extManager.SendI2CData(sendAddr, sendData, 3);
        if(ret == 0){
            MessageBox(TEXT("I2C Send Succeeded"));
        }else{ // Error
            MessageBox(TEXT("I2C Send Failed"));
        }
    }

}
```

送信が成功すると、以下のようなダイアログが表示されます。



- Receive I2C

接続されている SVM ボードに対し I2C データ受信要求を送信します。送信するデータやデバイスアドレス、受信されたデータを格納する配列は、TestDialog.cpp 内の OnBnClickedReceive() 関数内に定義されています。

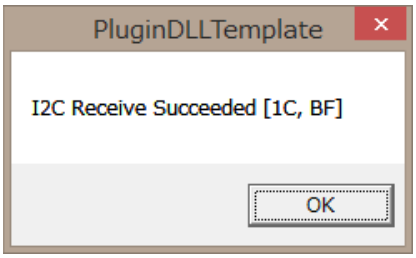
```
void CTestDialog::OnBnClickedReceive()
{
    // I2C データの受信
    if(m_pBaseFilter){
        CNVExtensionManager extManager(m_pBaseFilter);
        if(extManager.IsOpen() == 0){
            MessageBox(_T("Can't Get ExtensionUnit Interface."), _T("Error"), MB_OK);
            return;
        }

        unsigned char sendAddr = 0x3D; // Device addr
        unsigned char preambleData[1] = {0x20}; // preamble
        unsigned char receiveData[2];

        int ret = extManager.ReceiveI2CData(sendAddr, receiveData, 2, preambleData, 1);
        if(ret == 0){
            wchar_t text[64];
            wsprintf(text, TEXT("I2C Receive Succeeded [%02X, %02X]"), receiveData[0], receiveData[1]);
            MessageBox(text);
        }else{ // Error
            MessageBox(TEXT("I2C Receive Failed"));
        }
    }

}
```

受信が成功すると、以下のようなダイアログを表示します。この例では I2C データ [0x1C, 0xBF] が読まれたことを示しています。



3.4.4. 関数リファレンス

E	Ordinal ^	Hint	Function	Entry Point
C	1 (0x0001)	2 (0x0002)	ShowDialog	0x00001040
C	2 (0x0002)	0 (0x0000)	GetPluginName	0x00001120
C	3 (0x0003)	1 (0x0001)	GetPluginVersion	0x00001140

上図のように、DLL は 3 つの関数によって構成されます。

- ShowDialog

NVCap のプラグインメニューをクリックされたとき呼び出される関数です。通常、対応するモーダルダイアログを表示します。入力引数には、sPluginDLLData 構造体へのポインタが渡され、この内容を元に関数は DirectShow のインタフェースを生成することができます。

- GetPluginName

プラグインの名前を返す関数です。プラグインの名前は、NVCap のプラグインメニューに反映されます。

- GetPluginVersion

プラグインの対応するバージョンを返す関数です。現在のバージョンでは 100 を返してください。

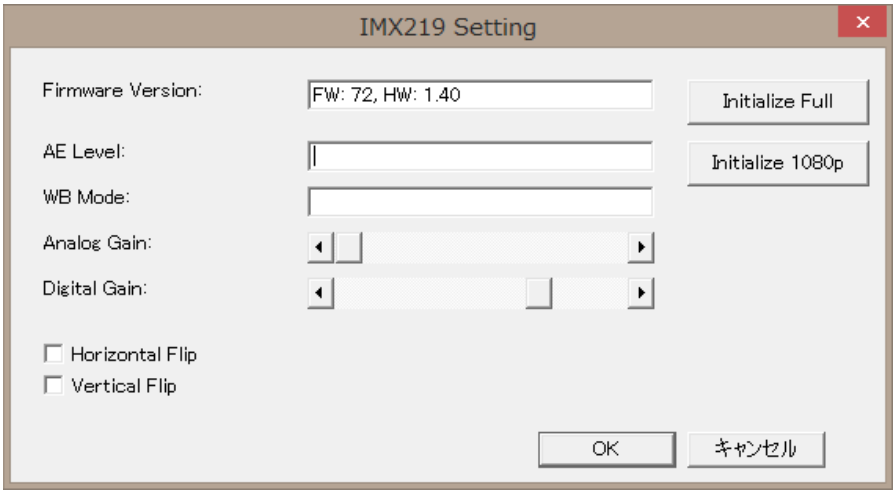
3.5. IMX219 用 PluginDLL プロジェクト

場所: SDK_Windows/PluginDLL_IMX219/

主要なファイル:

ファイル名	説明
PluginDLLTemplate.cpp PluginDLLTemplate.h	ダイアログ画面のインプリメントです。
TestDialog.cpp TestDialog.h	DLL のエクスポート関数のインプリメントです。
PluginDLLTemplate.def	DLL のエクスポート定義ファイルです。

3.5.1. 概要



「IMX219 用 PluginDLL プロジェクト」は、SVM-06 ボードに接続された Raspberry Pi Camera v2 のレジスタ設定を行うための NVCap 用プラグイン DLL プロジェクトです。上図のように、イメージセンサの Analog / Gain 調整や Flip を行うための GUI ダイアログによって構成されています。

3.6. NVFlipDLL

場所: SDK_Windows/NVFlipDLL/

主要なファイル:

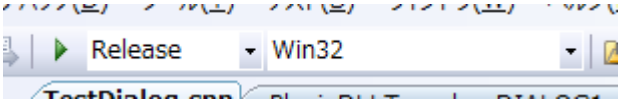
ファイル名	説明
FlipFilter.cpp FlipFilter.h	フィルタ部分の実装です。
NVFlipDLL.def	DLL のエクスポート定義ファイルです。

3.6.1. 概要

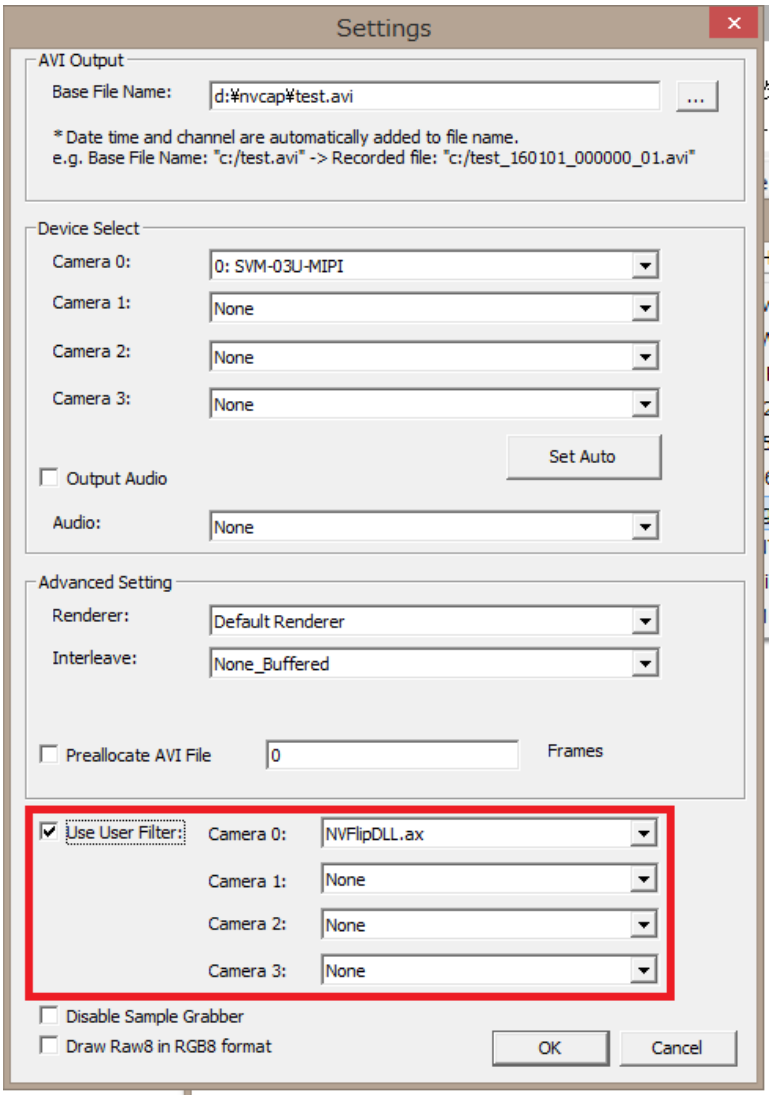
「NVFlipDLL」は NVCap の「UserFilter」機能のサンプルプロジェクトで、映像を上下反転するフィルタとして機能します。NVCap の設定で UserFilter を有効にして「NVFlipDLL」を選択することで、SVM ボードから入力された映像は上下反転して表示されます。UserFilter は .avi 形式への録画機能には影響しません。UserFilter をユーザが新規に実装する場合、このプロジェクトもしくは「NVRawDLL」プロジェクトをテンプレートとして使用してください。

3.6.2. ビルド方法

Visual Studio 2008 よりプロジェクトを開き、プロジェクトの構成を Release-Win32 もしくは Release - x64 に設定してビルドを行います。



Release-Win32 でビルドした場合「Release/NVFlipDLL.ax」Release-x64 でビルドした場合「x64/Release/NVFlipDLL.ax」が生成されます。この DLL ファイルを NVCap.exe（x64 の場合 NVCapx64.exe）が置かれたフォルダ内の「Filter」フォルダにコピーすることで、NVCap に UserFilter を登録します。x64 と x86 (Win32) のファイルを混同させると正常に動作しません。「Filter」フォルダ内の DLL ファイルは NVCap 起動時にスキャンされ、NVCap の Settings 画面内の User Filter リストに追加されます。



正常にビルドされた DLL を Filter フォルダにコピーした状態で NVCap を起動すると、上図 (NVCap のメニューより File - Settings) の赤枠内のリストボックスに「NVFlipDLL.ax」が追加されるはずです。これを選択して「Use User Filter」にチェックを入れることで、フィルタの効果が確認できます。

3.6.3. 動作詳細

UserFilter は DirectShow のフィルタとして実装され、フィルタ機能は CTransformFilter クラスから継承された CFlipFilter クラスの関数によって実装されます。

FilpFilter.cpp 内 MediaCheck() 関数によって、入力として受け入れるピクセルフォーマットを定義します。NVFlipDLL プロジェクトでは UYVY、YUY2 および RGB24、RGB8 フォーマットを受け入れるようにしています。SVM ボードが出力するピクセルフォーマットを

フィルタが受け入れない場合、NVCap のフィルタグラフが構築できず、正常にプレビューできないことがあります。

FlipFilter.cpp 内 Transform() 関数では、入力画像を受け取り出力画像に変換するフィルタとしての機能を定義します。NVFlipDLL プロジェクトでは入力画像をラインごとに分けて、上下反転して出力バッファにコピーしています。

NVFlipDLL をベースにして新たに UserFilter を製作する場合には、DLL のファイル名（*.ax）を変更して、NVCap の Filter フォルダに格納してください。このとき、FlipFilter.h 内の CLSID は変更しないでください。

NVFlipDLL では実装していませんが、UserFilter が ShowModelessDialog 関数をエクスポートすることで、フィルタ有効時にダイアログを表示することができます。実装例は「NVRawDLL」プロジェクトを参照してください。

3.7. NVRawDLL_MT / NVRawDLL

場所: SDK_Windows/NVRawDLL_MT/

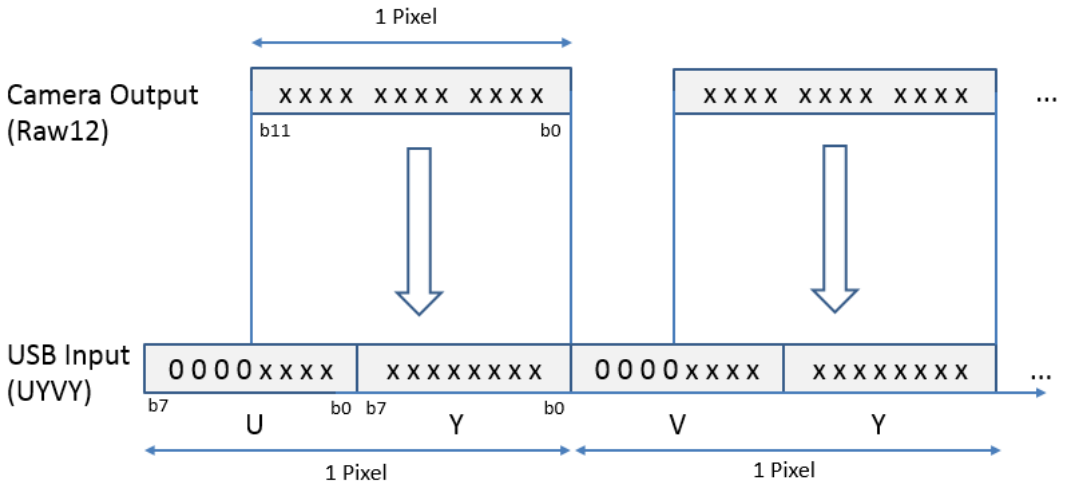
主要なファイル:

ファイル名	説明
FlipFilter.cpp FlipFilter.h	フィルタ部分の実装です。
NVFlipDLL.def	DLL のエクスポート定義ファイルです。
NVFlipDLL.cpp	エクスポート関数およびダイアログ部分の実装です。

3.7.1. 概要

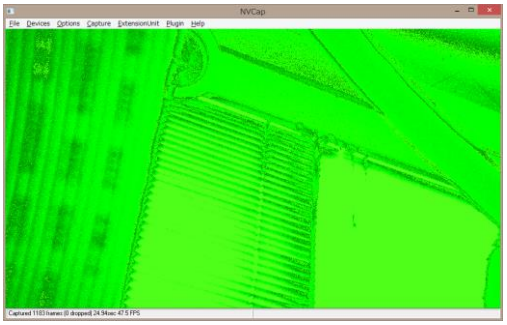
「NVRawDLL_MT」は NVCap の「UserFilter」機能のサンプルプロジェクトで、YUV 形式として入力される Raw フォーマットの映像をモノクロ表示したり、 デモザイク処理 (Bayer -> RGB 変換) してカラー表示を行うフィルタとして機能します。SVM-06 ボードで Raw 出力のイメージセンサを扱う場合は、この UserFilter を使用することで表示を行います。

「NVRawDLL」はシングルスレッド、「NVRawDLL_MT」はマルチスレッドであり、マルチスレッド版は処理が高速になっています。今後はマルチスレッド版のみ更新する予定です。

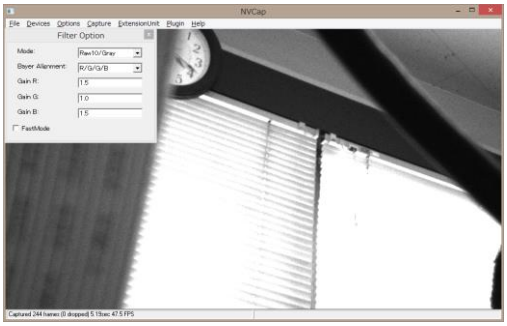


この UserFilter の取り扱う入力フォーマットは Raw 形式ですが、イメージタイプは UYVY として認識されており、上図 (Raw12 入力) のように各 Pixel の下位 bit に Raw10 / Raw12 フォーマットのデータがセットされます。UVC 規格では Raw10 / Raw12 入力を標準的にはサポートしないので、SVM ボードでは UYVY フォーマットとして出力する仕組みになっています。データとしては入力された映像データに変化を加えず、0 をパディングして出力する形になります。 NVRawDLL 挿入前後の映像例を下図に示します。

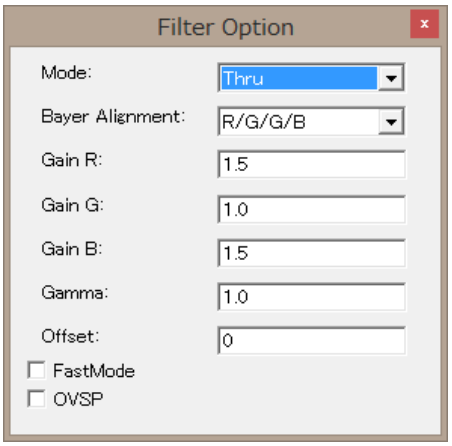
NVRawDLL 挿入前
(Raw 10bit を UYVY として取り込み)



NVRawDLL 挿入後
（Raw -> Gray 変換）



3.7.2. 動作詳細



前述の NVFlipDLL フィルタとは異なり、NVRawDLL_MT はダイアログによるフィルタ機能の変更に対応します。NVCap で NVRawDLL_MT を有効にすると、上図のような Filter Option ダイアログが表示されます。このダイアログの設定内容によってフィルタの機能や映像の調整を行います。

Mode: {Thru, Raw10/Gray, Raw10/Color, Raw12/Gray, Raw12/Color, ...}

- 有効なビット幅とグレースケール処理、カラー（デモザイク処理）を選択します。Thru のときはフィルタ処理を無効化します。

Bayer Alignment: {G/R/B/G, G/B/R/G, B/G/G/R, R/G/G/B}

- ベイヤー配列を設定します。これは Mode が /Color のときにのみ影響します。G/R/B/G の場合、映像左上から

G	R
B	G

の繰り返しとみなします。

Gain R / Gain G/ Gain B: RGB のゲインを設定します。

Gamma: ガンマカーブの係数を指定します。一般的なガンマ値とは異なる値となります。

Offset: 変換時、オフセットとして減算する値を指定します。

FastMode: チェックを入れると、表示が荒くなりますが処理が高速になります。/Color のときにのみ影響します。

OVSP: チェックを入れると、LSb/MSb を入れ替えます。

Filter Option ダイアログの実装は NVFlipDLL.cpp に記述されています。

3.8. SVMCtl_I2C

場所: SDK_Windows/SVMCtl_I2C/

3.8.1. 概要

SVMCtl は、SVM ボードを使用して I2C 通信や、SVM ボードのアップデート、センサの仕様設定、UVC 解像度設定等を行うための Windows 用ユーティリティソフトです。本 SDK に付属の「SVMCtl_I2C」は SVM ボードの CD に付属された標準版「SVMCtl」の I2C 通信部分のみを抜き出したプロジェクトです。