

SDK RELEASE NOTES

**VGA2USB,DVI2USB,
VGA2USB LR, VGA2USB HR, VGA2USB PRO,
DVI2USB SOLO, DVI2USB DUO
VGA2Ethernet
DVI2PCIe**

EPIPHAN SYSTEMS INC.

Introduction

この SDK には EpiphanSystems 社の VGA/DVI フレームグラバーのためのインターフェース定義ファイル (.h) が含まれています。

Quick Start

この SDK に含まれるサンプルは Microsoft Visual Studio 2005にて動作します。 SDK 中のソリューションファイルを開くと(SDK\epiphan\samples\v2u フォルダ)この SDK で提供されるすべてのプロジェクトが含まれています。

SDK の中核は”SDK\epiphan\frmgrab\include\frmgrab.h”ファイルで、Epiphan Systems のフレームグラバードライバーに支援される frmgrab ライブラリについて記述されています。

SDK\epiphan\samples\v2u_lib では、どの様に ioctls を使用するかが紹介されています。

SDK\epiphan\samples\v2u では、どの様に frmgrab API を利用するかが紹介されています。

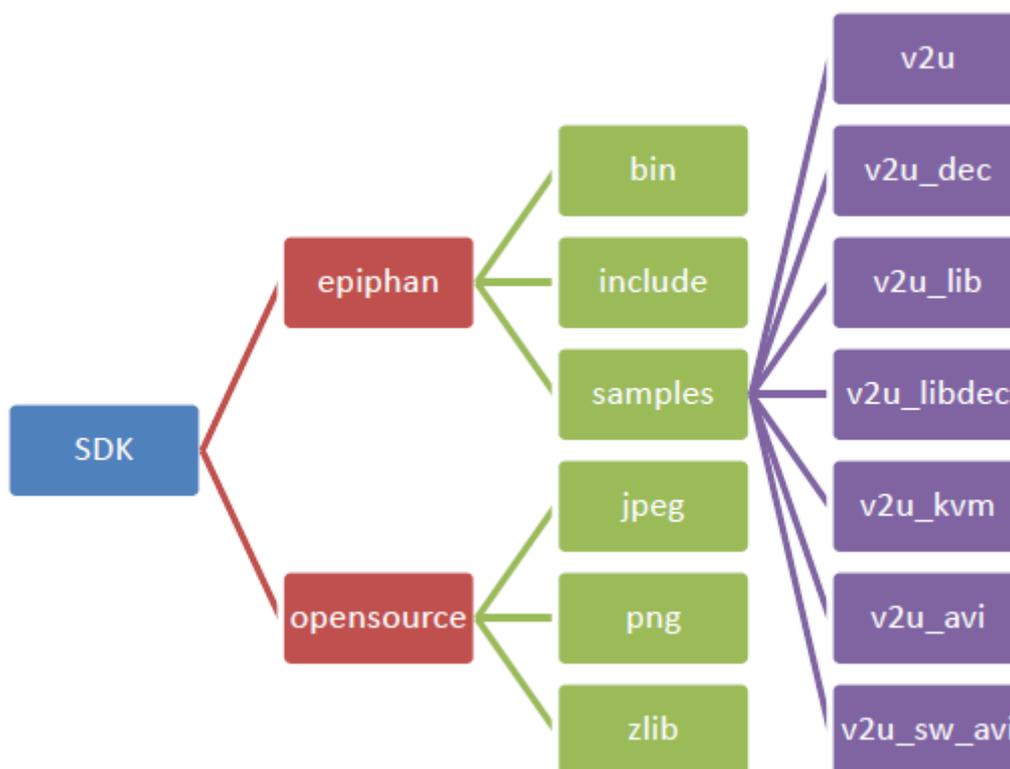
SDK\epiphan\samples\v2u_dec では、どの様に on-board 圧縮(raw format コンテナ)を利用するかを示しています。

SDK\epiphan\samples\v2u_libdec では、フレームを解凍するのに必要なdllを含んでいます。

SDK\epiphan\samples\v2u_avi では、on-board 圧縮されたフレームデータを AVI ファイルに格納する方法を紹介しています。(この AVI を再生するためには v2u_ds_decoder.ax が必要です。)

SDK Layout

SDK は下記の様に構成されています。



SDK\OPENSOURCE

このディレクトリにはキャプチャされたフレームを保存する OpenSource ライブラリが含まれます。

SDK\EPIPHAN\BIN

このディレクトリには予めコンパイルされたサンプルが用意されています。

- v2u.exe — キャプチャ、静止画保存、VGA モードの検知、コントラストやブライトネスなどのパラメータ設定など、簡単に操作を行うことのできるユーティリティです。
- v2u_dec.exe — EpiphanSystems の on-board 圧縮エンジンを使用したファイル保存やデコードを行うユーティリティです。引数に“-h”を指定することでヘルプが表示されます。このユーティリティは VGA2USB LR/ HR/ Pro と DVI2USB Duo/Solo でのみ動作します。
- v2u_libdec.dll — 解凍アルゴリズムライブラリ。v2u_dec.exe の動作に必要です。
- frmgrab.dll — Local 接続製品、ネットワーク製品にアクセスするために統合されたフレームグラバ API。v2u.exe に必要です。
- v2u_avi.exe — EpiphanSystems の on-board 圧縮エンジンを利用し、AVI ファイルに格納するためのユーティリティです。このユーティリティは VGA2USB LR/HR/PRO および、DVI2USB-Solo/Duo でのみ利用可能です。
- v2u_ds_decoder.ax — v2u_avi にて on-board 圧縮エンジンを利用して作成された AVI ファイルを再生させるための DirectShow コーデックです。3rd パーティの DirectShow 対応アプリケーションで利用できます。
- v2u_kvm.exe — KVM2USB にてマウスやキーボードを動作させるユーティリティです。

SDK\EPIPHAN\INCLUDE

このディレクトリには EpiphanSystems 社の VGA/DVI フレームグラバのためのインターフェース定義ファイル(.h)が含まれます。

SDK\EPIPHAN\FRMGRAB

このディレクトリには EpiphanSystems 社のネットワーク製品、および Local 接続製品に統合されたインターフェース定義ファイル(.h)が含まれます。このインターフェースを利用する Windows アプリケーションはランタイムに frmgrab.dll が必要になります。このライブラリは再配布可能です。Mac OS X および Linux 用のスタティックライブラリが含まれます。

SDK\EPIPHAN\SAMPLES

サンプルソースコードです。

on-board 圧縮(AVI FORMAT)

このセクションでは EpiphanSystems VGA2USB LR/HR/PRO および DVI2USB-Solo/Duo フレームグラバの on-board 圧縮エンジンを利用して AVI ファイルに直接書き出す方法を説明しています。このアプローチは画像取得プロセスで CPU 負荷を抑える事ができ、ローパフォーマンスな CPU プラットフォームや、CPU に負荷をかけたくないアプリケーションには最適なソリューションです。

RECORD AVI FILE

v2u_avi.exe ユーティリティはハードウェアの圧縮エンジンを利用し、そのまま AVI ファイルに保存します。このユーティリティは AVI ファイルの名前を引数とします。デフォルトでは RGB24色空間を使用してフレームを保存します。必要であれば、引数「-p」オプションを指定して YUV2色空間にて保存する事ができます。

例:

`v2u_avi.exe test.avi`

EPIPHAN デコーダ DirectShow Filter のインストール

EpiphanSystems VGA2USB LR/HR/PRO および DVI2USB-Solo/Duo フレームグラバードから v2u_avi.exe を利用して直接 AVI ファイルに保存したファイルを再生するためには、Epiphan Decoder DirectShow Filter をインストールする必要があります。

下記インストールプロセスです:

1. VGA フレームグラバードを接続しない状態で、これを利用するすべてのアプリケーションを閉じてください
2. 使用されていた古いバージョンのデコーダをアンインストールします

`regsvr32 /u <ここに以前のバージョンの v2u_ds_decoder.ax があるパスを指定します>`

3. 必要であれば、古いバージョンのデコーダを削除します
4. 新しいバージョンのデコーダをハードドライブに設置します

5. 新しい DirectShowFilter を登録します

`regsvr32 <v2u_ds_decoder.ax を設置したフルパスを指定します>`

AVI FILE の再生

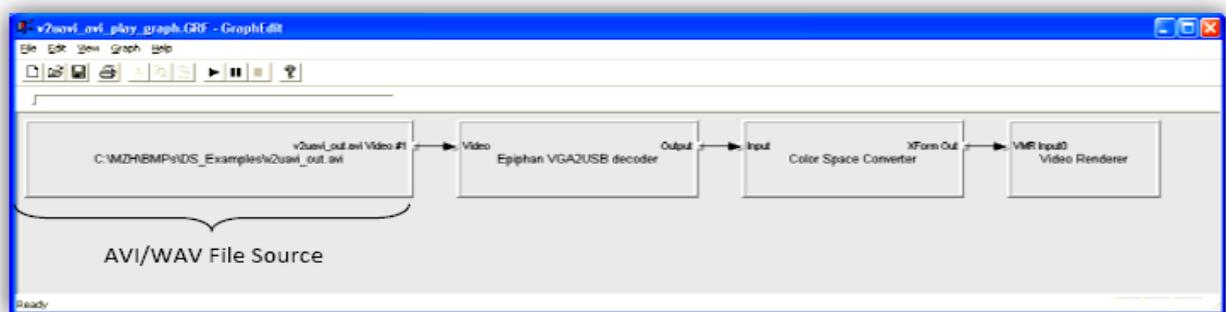
Microsoft Media Player

Epiphan Decoder DirectShow Filter をインストール後、Microsoft Media Player の様な DirectShow に対応したプレイヤーで Epiphan on-board 圧縮を利用した AVI ファイルを再生する事ができるようになります。

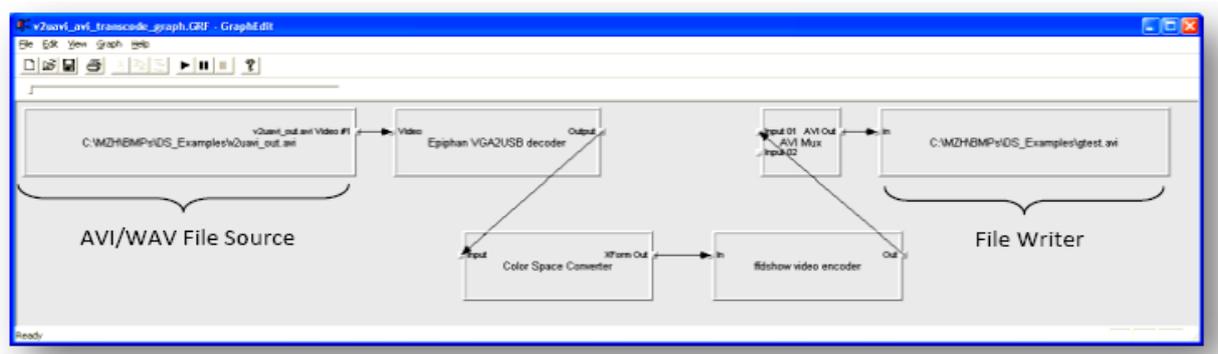
CUSTOM APPLICATIONS

Epiphan on-board 圧縮にて作成された AVI ファイルは以下のような DirectShow graph にて利用する事ができます。

再生:



トランスコーディング:



On-board 圧縮 (RAW FORMAT)

v2u_dec.exe ユーティリティは on-board 圧縮を使ったよりローレベルな働きをします。使用する前に下記を確認してください。

1. VGA2USB LR/HR/PRO もしくは DVI2USB-Solo/Duo フレームグラバラーのみが on-board 圧縮に対応しています。VGA2USB および DVI2USB ではこの機能を利用できません。
2. ドライババージョンが3.7.0.0000以上である必要があります。

圧縮されたフレームを save する

圧縮されたフレームをキャプチャするには、下記の様に v2u_dec.exe を利用します。

```
v2u_dec.exe 100 test.epm
```

これによりユーティリティは100フレームの画像データを test.epm ファイルにリニアに保存します。

保存されたフレームを解凍する

保存されたフレームを解凍するのも、v2u_dec.exe を利用します。

```
v2u_dec.exe x test.epm
```

ユーティリティは test.epm に格納されたフレームをすべて抽出し、次の名前を持ったファイルに保存します。

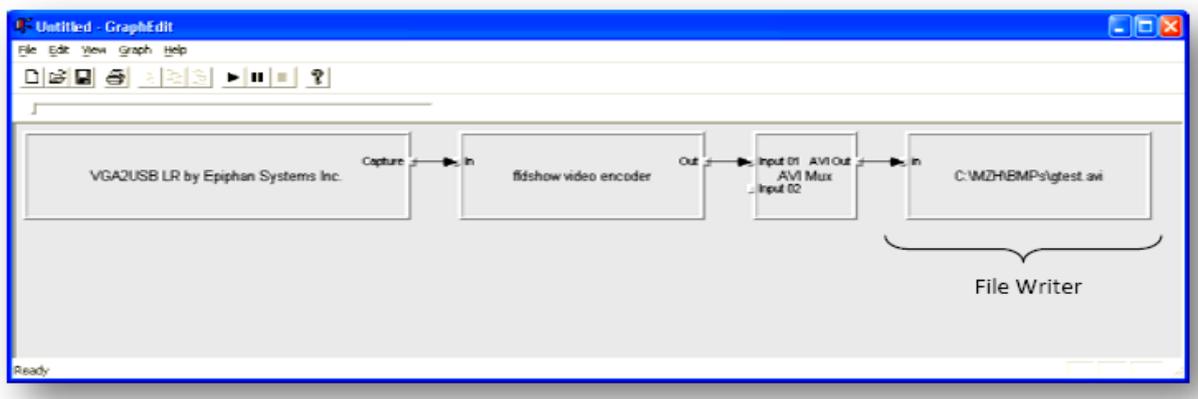
```
Test.epm.NNNN.bmp
```

NNNN はフレームのシーケンシャル番号となります。

ソフトウェア圧縮による AVI 保存

DirectShow を利用する

すべての Epiphan Systems のフレームグラバラーは DirectShow API をサポートしています。下にある graph は AVI ファイルへの保存を示しています。



v2u_dshow サンプルでは、どのようにデバイスを見つけ、フレームレートや解像度などの設定を行うかを示しています。

FRMGRAB API

Frmgrab は USB 製品とネットワーク製品にアクセスするための統合された API ライブラリです。このセクションでは epiphan\frmgrab\include\frmgrab.h に定義された FrmGrab API について解説します。

```
void FrmGrab_Init(void)
```

```
void FrmGrabNet_Init(void)
```

これら関数は FrmGrab ライブラリの内部データ構造を初期化します。しかし、Windows 上ではこれをコールする必要はありません。Windows バージョンの FrmGrab は DLL として提供されており、それがロードされると自動的に初期化されます。しかしながら、Max OS X や Linux 上で FrmGrab を利用するにはコールする必要があります。ネットワーク製品を利用するには FrmGrabNet_Init を、そうでなければ FrmGrab_Init をコールします。これら関数は他の FrmGrab 関数よりも先にコールされなければなりません。

void FrmGrab_Deinit(void) **void FrmGrabNet_Deinit(void)**

これら関数は FrmGrab_Init と FrmGrabNet_Init に対するもので、逆にデバイスの終了時に利用されます。同様に Windows ではコールする必要はありませんが、Mac OS X や Linux ではメモリの割り当てを削除するために終了時にコールする必要があります。

FrmGrabber* FrmGrab_Open(const char* location)

FrmGrab_Open は以下の文法を持った location パラメータによってフレームグラバードバイスを開きます。

| | |
|-----------------------|--|
| local: [SERIAL] | ローカルフレームグラバードバイスをシリアル番号で指定します。 |
| net: [ADDRESS[:PORT]] | address/port を特定し、ネットワークフレームグラバードバイスを指定します。もしアドレスが指定されない場合、ローカルネットワーク上のネットワークフレームグラバードバイスをランダムに開きます。 |
| sn: SERIAL | シリアル番号でネットワークフレームグラバードバイスを特定します。 ローカルフレームグラバードバイスをチェックし、その後ネットワークフレームグラバードバイスを探します。 |
| id: INDEX | インデックスを指定してローカルフレームグラバードバイスを特定します。 |

FrmGrabber* FrmGrab_Dup(FrmGrabber* fg)

FrmGrab_Dup 関数はフレームグラバードバイスのハンドルを2つに分けます。同じハードウェアを指す新しい独立した FrmGrabber インスタンスを返します。

const char* FrmGrab_GetSN(FrmGrabber* fg)

FrmGrab_GetSN 関数はフレームグラバードバイスのシリアル番号を返します。ポインタはフレームグラバードバイスのインスタンスが開いている限り有効です。

int FrmGrab_GetProductId(FrmGrabber* fg)

FrmGrab_GetProductId 関数はフレームグラバードバイスのユニークなプロダクト ID を返します。

const char* FrmGrab_GetProductName(FrmGrabber* fg)

FrmGrab_GetProductName 関数はモデル名の文字列を返します。(“VGA2USB”, “VGA2Ethernet” など)

const char* FrmGrab_GetLocation(FrmGrabber* fg)

FrmGrab_GetLocation 関数はデバイスの位置を文字列で返します。(“USB”, “192.168.0.122” など)

V2U_BOOL FrmGrab_DetectVideoMode(FrmGrabber* fg, V2U_VideoMode* vm)

FrmGrab_DetectVideoMode 関数はフレームグラバードバイスの検出されたビデオモードを返します。成功時には V2U_TRUE を、失敗時には V2U_FALSE を返します。vm パラメータが v2u_defs.h に定義された V2U_VideoMode 構造を指します。

```
typedef struct ioctl videomode {
    V2U_INT32 width;          /* 水平解像度、ピクセル */
    V2U_INT32 height;        /* 垂直解像度、ピクセル */
    V2U_INT32 vfreq;         /* 垂直リフレッシュレート mHz */
} V2U_VideoMode;
```

もし信号が検知されなかった場合、すべてのフィールドは0にセットされます。

```
/* ビデオモードを検知 */
V2U_VideoMode vm;
if (FrmGrab_DetectVideoMode(fg, &vm) && vm.width && vm.height) {

    printf("検知 %dx%d %d.%d Hz\n", vm.width, vm.height,
           (vm.vfreq+50)/1000, ((vm.vfreq+50)%1000)/100);

} else {
    printf("信号が検知されませんでした\n");
}
}
```

V2U_BOOL FrmGrab_GetGrabParams(FrmGrabber* fg, V2U_GrabParameters* gp)

FrmGrab_GetGrabParams 関数は現在の VGA キャプチャパラメータを問い合わせします。成功時には V2U_TRUE が、失敗時には V2U_FALSE が返されます。

V2U_BOOL FrmGrab_SetGrabParams(FrmGrabber* fg, const V2U_GrabParameters* gp)

FrmGrab_SetGrabParams 関数は VGA キャプチャパラメータをセットします。成功時には V2U_TRUE が、失敗時には V2U_FALSE が返されます。

V2U_BOOL FrmGrab_GetProperty(FrmGrabber* fg, V2U_Property* prop)

FrmGrab_GetProperty 関数はデバイスプロパティを問い合わせます。成功時には V2U_TRUE が、失敗時には V2U_FALSE が返されます。これにより v2u_defs.h に定義された V2UPropertyKey enum 値の一つに prop->key フィールドをセットします。成功が返された場合、プロパティの値は Prop->value に見ることができます。

```
/* KVM 機能がサポートされているかのチェック */
V2U_Property p;
p.key = V2UKey_KVMCapable;
if (FrmGrab_GetProperty(fg, &p)) {
    if (p.value.boolean) {
        ... // 実行部
    } else {
        printf("フレームグラバは KVM 機能をサポートしていません\n");
    }
}
}
```

V2U_BOOL FrmGrab_SetProperty(FrmGrabber* fg, const V2U_Property* prop)

FrmGrab_SetProperty 関数はデバイスのプロパティをセットします。成功時には V2U_TRUE が、失敗時には V2U_FALSE が返されます。

V2U_BOOL FrmGrab_SendPS2(FrmGrabber* fg, const V2U_SendPS2* ps2)

FrmGrab_SendPS2関数は KVM 機器向けのコマンドです。

void FrmGrab_Start(FrmGrabber* fg)

FrmGrab_Start 関数はフレームグラバーに対して最高フレームレートにてフレームキャプチャするための準備をする合図を出します。現在この関数はローカルのフレームグラバーには重要ではありませんが、ネットワーク製品には非常に重要です。ネットワーク製品にはこの関数によりストリーミングを実行状態にしておきます。これを怠ると、FrmGrab_Frame は リクエスト-反応 といった動作となり、遅くなってしまいます。

void FrmGrab_Stop(FrmGrabber* fg)

FrmGrab_Stop は最高フレームレートでキャプチャしている動作の準備を停止します。

V2U_BOOL FrmGrab_SetMaxFps(FrmGrabber* fg, double maxFps)

FrmGrab_SetMaxFps 関数はフレームレート制限をセットします(秒間あたりの平均値)。これにより、リソースの使用量を制限することが可能です。例えばネットワーク製品にこれを利用した場合、ネットワーク帯域を制限することが可能になります。

V2U_GrabFrame2* FrmGrab_Frame(FrmGrabber* fg, V2U_UINT32 format, const V2URect* crop)

FrmGrab_Frame 関数は一枚のフレームを取得します。これは FrmGrab_Start を予めコールする必要はありませんが高速な反応を得るために予めコールしておくことが推奨されます。2番目のパラメータはキャプチャフォーマットです。これは v2u_defs.h に定義された V2U_GRABFRAME_FORMAT_*定数のうちの一つです。最後のパラメータはリクエストされた長方形を切り出すためのポインターです。キャプチャフレームのすべての部分が必要な場合は NULL としてください。これにより取得したフレームが必要ではなくなった際には必ず FrmGrab_Release をコールしてフレームをリリースする必要があります。

void FrmGrab_Release(FrmGrabber* fg, V2U_GrabFrame2* frame)

FrmGrab_Release 関数は FrmGrab_Frame によって返された以前のフレームをリリースします。

void FrmGrab_Close(FrmGrabber* fg)

FrmGrab_Close 関数はフレームグラバーを閉じ、ハンドルを無効にします。FrmGrab_Frame によって返されたフレームはすべて FrmGrab_Close をコールする前にリリースする必要があります。

USB製品専用関数

FrmGrabber* FrmGrabLocal_Open(void)

FrmGrabLocal_Open 関数はデフォルトの USB フレームグラバーを開きます。もし複数のフレームグラバーが接続されている場合、どのユニットが開かれるか選定できません。フレームグラバーがコンピュータ上に見つからない場合、NULL を返します。

FrmGrabber* FrmGrabLocal_OpenSN(const char* sn)

FrmGrabberLocal_OpenSN 関数は特定のシリアル番号の USB フレームグラバーを開きます。リクエストされたシリアル番号のユニットが見つからない場合は NULL を返します。

int FrmGrabLocal_Count(void)

この関数はシステムに接続されている USB フレームグラバーの数を返します。

int FrmGrabLocal_OpenAll(FrmGrabber* grabbers[], int maxcount)

FrmGrabLocal_OpenAll は複数の USB フレームグラバーを一度に開きます。実際に開かれたフレームグラバーの数を返します。

ネットワーク製品専用関数

FrmGrabber* FrmGrabNet_Open(void)

FrmGrabNet_Open 関数はサブネット上にあるネットワークグラバー製品を見つけ、開こうと試みます。成功時にはフレームグラバーのハンドルを返し、失敗時には NULL を返します。

FrmGrabber* FrmGrabNet_OpenSN(const char* sn)

FrmGrabNet_OpenSN 関数は特定のシリアル番号のネットワークグラバー製品を見つけ、開こうと試みます。

FrmGrabber* FrmGrabNet_OpenLocation(const char* location)

FrmGrabNet_OpenLocation 関数は特定の位置 (ホストネームか IP アドレス) にあるフレームグラバーに接続します。

FrmGrabber* FrmGrabNet_OpenAddress(V2U_UINT32 ipaddr, V2U_UINT16 port)

FrmGrabNet_OpenAddress 関数は特定の IP アドレスにあるフレームグラバーに接続します。接続するためのデフォルトポートとしてポート0が送られます。

FrmGrabber* FrmGrabNet_OpenAddress2(V2U_UINT32 ipaddr, V2U_UINT16 port, FrmGrabAuthProc authproc, void* param, FrmGrabConnects* status)

この関数は特定の IP アドレスにあるネットワークフレームグラバーに接続します。接続するためのデフォルトポートとしてポート0が送られます。フレームグラバーが認証を必要とする場合、ユーザー名とパスワードを得るために、FrmGrabAutoProc コールバックが呼ばれます。FrmGrabAuthProc コールバックは以下の形となります：

```
typedef V2U_BOOL (*FrmGrabAutoProc) (char* user, char* pass, void* param);
```

ユーザー名の最大サイズは FG_USERNAME_SIZE(32) で、パスワードの最大サイズは FG_PASSWORD_SIZE(64) bytes です。ユーザー名もパスワードも UTF-8 にてエンコードされている必要があります。パスワードは決してネットワーク上に送られません。ユーザー名が必要無い場合、パスワードは NULL です。

これは FrmGrabNet_OpenAddress と必要な場合に FrmGrabNet_Auth 機能を統合した便利な関数です。

V2U_BOOL FrmGrabNet_IsProtected(FrmGrabber* fg)

FrmGrabNet_IsProtected 関数はネットワークフレームグラバーがパスワード保護されているかをチェックします。FrmGrabNet_Open や FrmGrabNet_OpenSN、FrmGrabNet_OpenLocation 関数は認証がないクライアントにハンドルを返すかもしれません。これを防ぐために、予め認証を確認する必要があります。これらは FrmGrabNet_Auth 機能の下で使用される必要があります。

**FrmGrabConnectStatus FrmGrabNet_Auth(FrmGrabber* fg,
FrmGrabAuthProc authproc, void* param)**

FrmGrabNet_Auth は必要な際にクライアントを判定します。FrmGrabNet_OpenAddress2関数を参照してください。

V2U_BOOL FrmGrabNet_GetStat(FrmGrabber* fg, FrmGrabNetStat* netstat)

FrmGrabNet_GetStat 関数はネットワークへの送出バイト数や受信バイト数など、ネットワーク統計を返します。

V2U_BOOL FrmGrabNet_GetRemoteAddr(FrmGrabber* fg, struct sockaddr_in* addr)

FrmGrabNet_GetRemoteAddr 関数はフレームグラバーのアドレスを返します。