NM33 用

OptCamSDK
取扱説明書

Rev. 1.05　　2018/01/17

オプト株式会社　　OPT Corporation
〒391-0013 長野県茅野市宮川 5423-2　　Tel: 0266-82-0020
5423-2 Miyagawa,Chino-shi,Nagano-ken 391-0013, Japan　　Fax: 0266-82-0022

目　次

1. 概要

OptCamSDK は、ユーザ作成アプリケーションとリンクし、NM33 カメラの主要機能の操作が可能なソフトウェア開発キットです。

本取扱説明書は、 OPT 標準ドライバ版 F/W （以下、OPT 版） 向け DLL と、Windows 標準ドライバ版 F/W （以下、UVC 版） 向け DLL の両方でお使いいただけます。

本書の対象 DLL バージョン

OPT 版 DLL …v0.012 以降

UVC 版 DLL …v.1.012 以降

2. 機能

- 接続されているカメラの認識と使用カメラの選択
- カメラ撮影画像の取得
- カメラ撮影位置や撮影条件の取得/設定
- 撮影時情報表示の設定(有効/無効)
- カメラ設定の初期化実行

3. 構成

- SSK.dll・・・DLL
- SSK.lib・・・アプリケーション作成用ライブラリ
- SSK.h ・・・アプリケーション作成用関数ヘッダ

4. 動作環境

OS: Windows Vista、Windows 7, 8, 8.1 および 10

PC:上記 OS が正常に動作する環境

(推奨スペック)

CPU: Pentium 3 以上(Pentium 4 /M/D/Dual-Core,Core Solo/Duo 等)

メモリ: 256 Mbyte (Vista の場合は 768Mbyte) 以上

HDD: 20Gbyte 以上

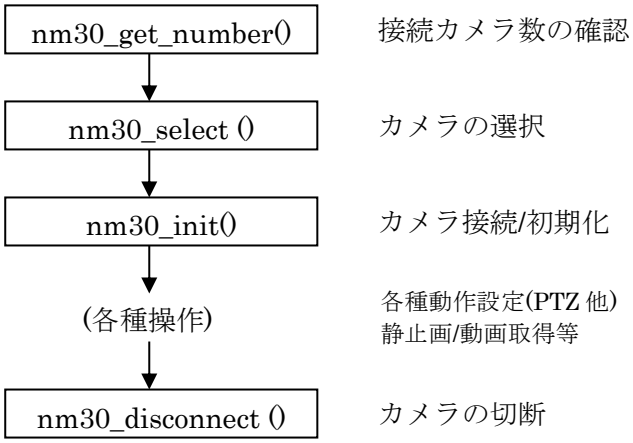
5. 関数の使用方法

- 使用環境

  本 DLL では、USB を介して NM33 を操作します。
  NM33 は、RS232C による通信をサポートしていますが、本 DLL では RS232C は使用しません。

- デバイスの確認/選択と初期化～終了

  接続されているカメラ数の確認と使用カメラを選択後、接続/初期化を行います。
  接続/初期化が正常に完了すると、各種設定や静止画/動画の取得が可能になります。
  アプリ終了時にはカメラの切断を行い、デバイスとの接続を終了します。



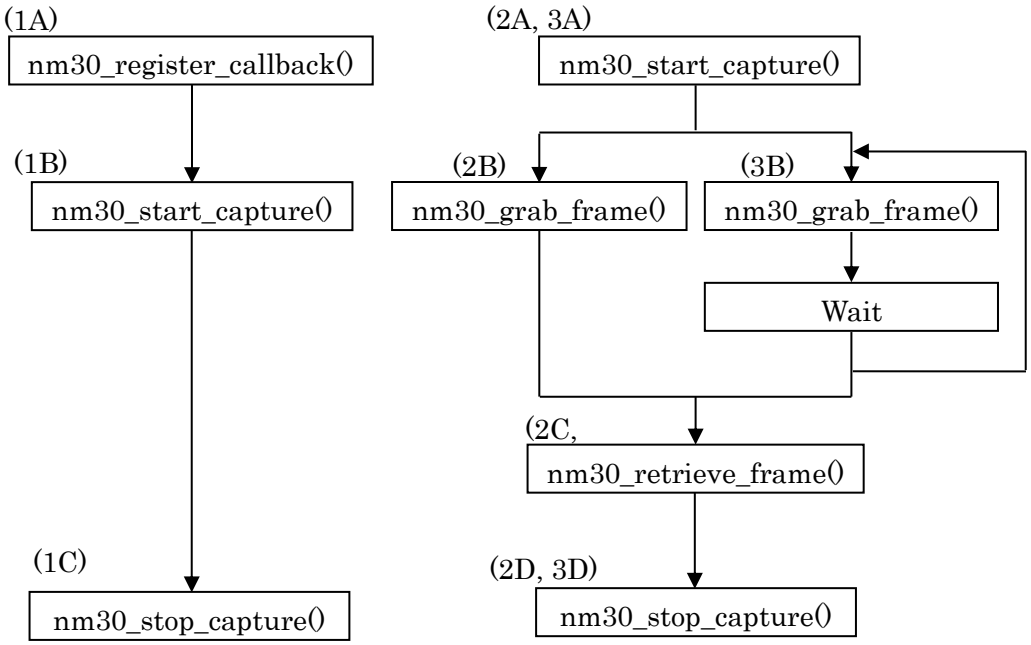
- 映像フレームの取得

カメラから映像フレームを取得する方法は、アプリケーションの作成形式に応じて３タイプの取得方法が選択できます。

1. コールバック関数を登録してキャプチャ終了後に呼び出されるようにする方法
   (A)コールバック関数登録→(B)キャプチャ開始→(C)キャプチャ停止 の順番で関数を呼び出します。

2. キャプチャ関数を呼んでフレームが取得できるまで待つ方法
   (A)　　キャプチャ開始→(B)映像フレーム受信確認→(C)映像フレーム取得→(D)キャプチャ停止 の順番で関数を呼び出します。

3. キャプチャ関数をポーリングモードで呼んでフレームが取得できるまで待つ方法
   (A)キャプチャ開始→(B)映像フレーム受信確認→・・・→(B)映像フレーム受信確認→(C) 映像フレーム取得→(D)キャプチャ停止 の順番で関数を呼び出します。

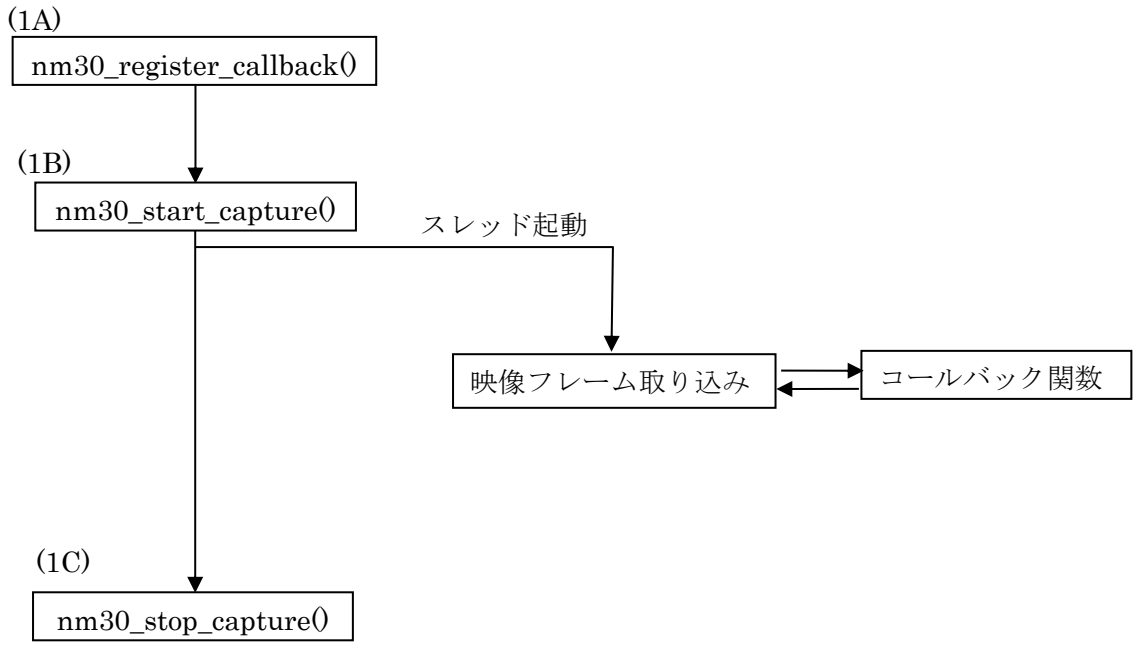


(タイプ別詳細)

タイプ１では、nm30_register_callback()でコールバック関数を登録した後、nm30_start_capture()を実行します。

nm30_start_capture()により、映像フレーム取り込み専用のスレッドが起動されます。

1 枚の映像フレームが取り込まれたら、登録したコールバック関数が呼ばれます。



コールバック関数の処理が完了後、制御は DLL 内部の映像フレーム取り込み部に戻り、引き続き、次の映像フレームの取り込みが開始されます。これは、nm30_stop_capture()でキャプチャを停止するまで繰り返されます。

タイプ１では、nm30_grab_frame()、nm30_retrieve_frame()は使用できません。

タイプ２、３で行う場合は、nm30_start_capture()実行後に、nm30_grab_frame()、nm30_retrieve_frame()を使用します。タイプ２とタイプ３は、nm30_grab_frame()へ渡す引数の値の違いで決まります。

nm30_retrieve_frame()で映像フレームを取得した後、引き続き、nm30_grab_frame()を実行することで、次の映像フレームを取り込むことができます。

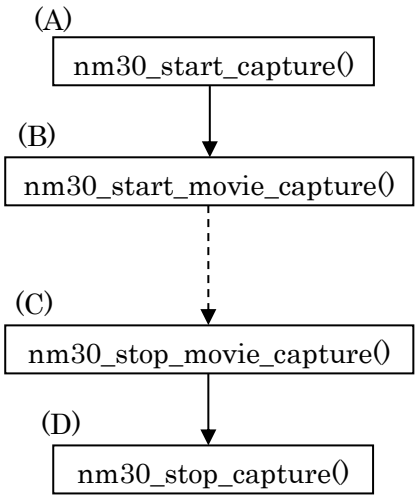
nm30_stop_capture()は映像フレーム一枚ごとに使用する必要はありません。

- 動画のキャプチャ

  動画のキャプチャはファイルへ直接出力を行います。

  (A)　　キャプチャ開始→(B)動画出力開始→(C)動画出力終了

  →(D)キャプチャ停止 の順番で関数を呼び出します。



- Large サイズ(1536x1536)画像について

  Large サイズ画像については、キャプチャ画像サイズ指定とパノラマ映像の種類が連動して切り替わります。

  (設定/取得関数)

  パノラマ映像の種類：　nm30_set/get_panorama_mode()

  キャプチャ画像サイズ：nm30_set_capture_size()

  nm30_get_capture_width()/nm30_get_capture_height()

  ・nm30_set_capture_size()で 1536x1536 を指定した場合

  パノラマ映像の種類として 9(円形(Large サイズ))を指定した状態になります。

  ・nm30_set_capture_size()で 1536x1536 以外を指定した場合

  パノラマ映像の種類は Large サイズ移行前の画面モードに変更されます。

  起動時に Large サイズだった場合は、0(円形)に変更されます。(OPT 版のみ。UVC 版では Large サイズで起動させることができないため、本動作は適用しません)

  ・nm30_set_panorama_mode()で 9 (円形(Large サイズ))以外を設定した場合

  キャプチャ画像サイズは自動的に VGA サイズに変更されます。

  VGA サイズ以外で表示したい場合、nm30_set_capture_size()で希望サイズを指定してください。

6. 関数一覧

表 1　関数一覧

関数	機能
nm30_init()	カメラの初期化を行う
nm30_get_number()	接続されているカメラの数を取得する
nm30_get_id()	指定したカメラの ID を取得する
nm30_param_save()	カメラに設定値を保存する
nm30_select()	カメラを選択する
nm30_set_panorama_mode()	パノラマ映像の種類を切り替える
nm30_get_panorama_mode()	現在のパノラマ映像の種類を取得する
nm30_set_pan()	パン値を設定する
nm30_get_pan()	現在のパン値を取得する
nm30_set_tilt()	チルト値を設定する
nm30_get_tilt()	現在のチルト値を取得する
nm30_set_zoom()	ズーム値を設定する
nm30_get_zoom()	現在のズーム値を取得する
nm30_set_roll()	ロール値を設定する
nm30_get_roll()	現在のロール値を取得する
nm30_enable_information_display()	フレームレート等の情報を表示する
nm30_disable_information_display()	フレームレート等の情報を非表示にする
nm30_enable_overlay_display()	魚眼映像のオーバーレイ表示を有効にする
nm30_disable_overlay_display()	魚眼映像のオーバーレイ表示を無効にする
nm30_jpeg_get()	現在の JPEG 圧縮率を取得する
nm30_jpeg_set()	JPEG 圧縮率を設定する
nm30_set_autopan_speed()	オートパン速度を設定する
nm30_get_autopan_speed()	現在のオートパン速度を取得する
nm30_set_flip_screen()	フリップ/ミラー状態を設定する
nm30_get_flip_screen()	現在のフリップ/ミラー状態を取得する
nm30_set_sharpness()	シャープネスフィルタを設定する
nm30_get_sharpness()	現在のシャープネスフィルタを取得する
nm30_set_exposure_time()	シャッタースピードを設定する
nm30_get_exposure_time()	現在のシャッタースピードを取得する
nm30_set_gain()	ゲイン値を設定する
nm30_get_gain()	現在のゲイン値を取得する
nm30_enable_auto_exposure()	AE(自動露光)を有効にする
nm30_disable_auto_exposure()	AE(自動露光)を無効にする

nm30_set_capture_size()	映像キャプチャ時の幅と高さを設定する
nm30_get_capture_width()	現在の映像キャプチャ時の幅を取得する
nm30_get_capture_height()	現在の映像キャプチャ時の高さを取得する
nm30_set_capture_fps()	フレームレートを設定する
nm30_get_capture_fps()	現在のフレームレートの設定値を取得する
nm30_get_actual_fps()	現在の実効フレームレートを取得する
nm30_start_capture()	キャプチャを開始する
nm30_register_callback()	キャプチャ時のコールバック関数を登録する
nm30_grab_frame()	映像フレームをキャプチャする
nm30_retrieve_frame()	映像フレームを取得する
nm30_start_movie_capture()	動画キャプチャを開始する
nm30_stop_movie_capture()	動画キャプチャを終了する
nm30_stop_capture()	キャプチャを終了する
nm30_disconnect()	カメラとの接続を切断する

7. 関数

- int nm30_init (void)

  カメラ との接続を確認し、初期化を行う。

  戻り値　：0 正常終了、1 接続エラー、2 初期化エラー

- int nm30_get_number (void)

  現在コンピュータに接続されているカメラの数を取得する。

  戻り値　：接続されているカメラの数

  備考　　：OPT 版の場合、1 台のみ接続可能のため、戻り値は 0 か 1 のみ。

- char* nm30_get_id (int id)

  カメラ固有の id を取得する。

  引数　　：カメラの番号(1~)

  戻り値　：カメラ固有の id(例:NM33-012345) が格納された char 型配列へのポインタ

- int nm30_param_save (int id)

  カメラの現在の設定値を保存する。

  引数　　：カメラの番号(1~)

  戻り値　：0 正常終了、1 エラー

- int nm30_select (int id)

  カメラを選択する。

  引数　　：カメラの番号(1~)

  戻り値　：0 正常終了、1 エラー

  備考　　：これ以降に示される関数による操作はここで選択されたカメラに適用されます。

- int nm30_set_panorama_mode (int mode)

  パノラマ映像の種類を切り替える。

  引数　　：0 円形、1 周回、2 周回(反転)、3 上下二段パノラマ、
  4 上下二段パノラマ（反転)、5 ４画面、6 ４画面(反転)
  9 円形(Large サイズ)、11 直交一画面

  戻り値　：0 正常終了、1 エラー

  ※本関数では反転/非反転の状態を flip の設定により切り替えており、
  フリップ/ミラー状態(nm30_set_flip_screen())の設定値を更新します。

- int nm30_get_panorama_mode (void)
  現在のパノラマ映像の種類を取得する。
  戻り値　：0 円形、1 周回、2 周回(反転)、3 上下二段パノラマ、
  4 上下二段パノラマ（反転)、5 4 画面、6 4 画面(反転)
  9 円形(Large サイズ)、11 直交一画面

  ※本関数では flip の設定値により反転/非反転の状態を判別しているため、パノラマ映像種類の設定後、フリップ/ミラー状態(nm30_set_flip_screen())を変更した場合、本関数の戻り値がパノラマ映像種類の設定時と異なる場合があります。

- int nm30_set_pan (float pan)
  パノラマ映像のパンを設定する。
  引数　：パン角度 [deg]
  戻り値　：0 正常終了、1 エラー

- float nm30_get_pan (void)
  現在のパノラマ映像のパンを取得する。
  戻り値　：パン角度 [deg]

- int nm30_set_tilt (float tilt)
  パノラマ映像のチルトを設定する。
  引数　：チルト角度 [deg]
  戻り値　：0 正常終了、1 エラー

- float nm30_get_tilt (void)
  現在のパノラマ映像のチルトを取得する。
  戻り値　：チルト角度 [deg]

- int nm30_set_zoom (float zoom)
  パノラマ映像のズームを設定する。
  引数　：OPT 版･･･ズーム角度 [deg]
  UVC 版･･･ズーム倍率
  戻り値　：0 正常終了、1 エラー

- float nm30_get_zoom (void)
  現在のパノラマ映像のズームを取得する。
  戻り値　：OPT 版･･･ズーム角度 [deg]
  UVC 版･･･ズーム倍率

- int nm30_set_roll (float roll)
  パノラマ映像のロールを設定する。
  引数　　：ロール角度 [deg]
  戻り値　：0 正常終了、1 エラー

- float nm30_get_roll (void)
  現在のパノラマ映像のロールを取得する。
  戻り値　：ロール角度 [deg]

- int nm30_enable_information_display (void)
  フレームレート等の情報表示を有効にする。
  戻り値　：0 正常終了、1 エラー
  備考　　：情報は、画像上にオーバーレイで表示されます。

- int nm30_disable_information_display (void)
  フレームレート等の情報表示を無効にする。
  戻り値　：0 正常終了、1 エラー

- int nm30_enable_overlay_display (void)
  魚眼映像のオーバーレイを有効にする。
  戻り値　：0 正常終了、1 エラー

- int nm30_disable_overlay_display (void)
  魚眼映像のオーバーレイを無効にする。
  戻り値　：0 正常終了、1 エラー

- int nm30_jpeg_get(void)
  現在の JPEG 圧縮率を取得する。
  戻り値　：JPEG 圧縮率(０～９９の整数)

- int nm30_jpeg_set(int jpegcomp)
  JPEG 圧縮率を設定する。
  引数　　：JPEG 圧縮率(０～９９の整数)
  戻り値　：0 正常終了、1 エラー

- int nm30_set_autopan_speed (int speed)
  オートパン速度を設定する
  引数　　：オートパン速度(-3～3)
  戻り値　：0 正常終了、1 エラー

- int nm30_get_autopan_speed(void)
  現在のオートパン速度を取得する
  戻り値　：オートパン速度

- int nm30_set_flip_screen (int mode)
  フリップ/ミラー状態を設定する
  引数　　：フリップ/ミラー状態
  　　0:反転なし
  　　1:水平反転
  　　2:垂直反転
  　　3:水平＋垂直反転
  戻り値　：0 正常終了、1 エラー
  ※本項目はパノラマ映像種類の設定(nm30_set_panorama_mode())により、
  　設定値が更新される場合があります。

- int nm30_get_flip_screen(void)
  現在のフリップ状態を取得する
  戻り値　：フリップ状態
  ※本項目はパノラマ映像種類の設定(nm30_set_panorama_mode())により、
  　設定値が更新される場合があります。

- int nm30_set_sharpness (int filter)
  シャープネスフィルタを設定する
  引数　　：フィルタ番号(0～8)
  戻り値　：0 正常終了、1 エラー

- int nm30_get_sharpness(void)
  現在のシャープネスフィルタを取得する
  戻り値　：フィルタ番号

- int nm30_set_exposure_time (int time)
  シャッタースピード(露光時間)を設定する
  引数　　：OPT 版･･･シャッタースピード(5～500000μs)
  　　UVC 版･･･シャッタースピード(露光時間を 2^n で指定、-10～-1)
  戻り値　：0 正常終了、1 エラー

- int nm30_get_exposure_time(void)
  現在のシャッタースピード(露光時間)を取得する
  戻り値　：シャッタースピード

- int nm30_set_gain (int gain)
  ゲインを設定する
  引数　　：ゲイン値 x1000(10～32000)
  戻り値　：0 正常終了、1 エラー

- int nm30_get_gain(void)
  現在のゲインを取得する
  戻り値　：ゲイン値 x1000

- int nm30_enable_auto_exposure (void)
  AE(自動露光)を有効にする
  戻り値　：0 正常終了、1 エラー

- int nm30_disable_auto_exposure (void)
  AE(自動露光)を無効にする
  戻り値　：0 正常終了、1 エラー

- int nm30_set_capture_size (int width, int height)
  映像キャプチャ時の幅と高さを設定する。
  引数　　：OPT 版･･･
  幅（ピクセル:112~640(16 の倍数)、1536）
  高さ（ピクセル:2~480(2 の倍数)、1536）

  UVC 版･･･
  下記数値の組み合わせを指定します。

サイズ	幅(width)	高さ(height)
QVGA	320	240
VGA	640	480
LARGE	1536	1536

  戻り値　：0 正常終了、1 エラー

- int nm30_get_capture_width (void)
  現在の映像キャプチャ時の幅を取得する。
  戻り値　：幅（ピクセル）

- int nm30_get_capture_height (void)
  現在の映像キャプチャ時の高さを取得する。
  戻り値　：高さ（ピクセル）

- int nm30_set_capture_fps (float fps)
  映像キャプチャ時の fps(フレーム/秒)を設定する。
  引数　　：fps(0, 6.4～16.0)
  　　0 は自動フレームレート制御に切替。
  　　0 以外では、固定フレームレート制御に切替。
  戻り値　：0 正常終了、1 エラー

- float nm30_get_capture_fps (void)
  現在の映像キャプチャ時の fps(フレーム/秒)の設定値を取得する。
  戻り値　：fps
  　　0 の場合は自動フレームレート制御。
  　　0 以外の場合は、固定フレームレート制御で、現在の設定値。

- float nm30_get_actual_fps (void)
  現在の映像キャプチャ時の実効 fps を取得する。
  戻り値　：fps

- int nm30_start_capture (void)
  映像のフレームバッファへのキャプチャを開始する
  戻り値　：0 正常終了、1 エラー

- nm30_register_callback(ImagecallbackType)
  映像フレームのキャプチャ終了後に呼ばれるコールバックを登録する。
  引数　　：コールバック関数へのポインタ(SSK.h で定義)
  戻り値　：0 設定失敗、1 設定成功

  ・コールバック関数の形式

```
typedef INT (nudecl *Imagecallbacktype)
  (INT nSize,              :映像フレームのサイズ
   UCHAR* pData,           :映像フレームデータのポインタ
   USHORT Width,           :映像フレームの幅
```

```
    USHORT Height,          :映像フレームの高さ
    USHORT Format           :映像フレームの種類(JPEG 固定)
    SYSTEMTIME Time);       :映像フレーム取得時刻
```

- int nm30_grab_frame (int mode)

  映像フレームをキャプチャする。

  引数　　：0　フレームが取得できるまで待つ
  　　1　フレームが取得できるまで待たない（ポーリングモード）

  戻り値　：0 取得失敗、1 取得成功

  備考　　：映像フレームは DLL 内部で確保したメモリ領域に格納されます。

- ImageStract* nm30_retrieve_frame (void)

  取得した映像フレームへのポインタを返す

  戻り値　：映像フレームへのポインタ(SSK.h で定義)

  ・映像フレーム情報構造体定義

```
typedef struct {
  BYTE*   StreamImageBytes;     :映像フレームのポインタ
  long    StreamLength;         :映像フレームのサイズ
  long    ImageWidth;           :映像フレームの幅
  long    ImageHeight;          :映像フレームの高さ
  SYSTEMTIME Time;              :映像フレーム取得時刻
} ImageStruct;
```

  備考　　：StreamImageBytes には、JPEG ヘッダを含む JPEG 画像データ全体が格納されていますので、そのまま JPEG データとして使用可能です。
  取得したポインタは、DLL 側で解放しますので、アプリケーション側で解放しないでください。

- int nm30_start_movie_capture (char* filepath, int filetype)

  動画キャプチャを開始する

  引数　　：filepath：出力ファイルのパス(フルパス)
  　　filetype：出力ファイルの種別
  　　0(TYPE_AVI)：AVI 形式
  　　1(TYPE_MJPEG)：MJPEG 形式

  戻り値　：0 正常終了、1 エラー

- int nm30_stop_movie_capture (void)
  動画キャプチャを終了する
  引数　　：なし
  戻り値　：0 正常終了、1 エラー

- int nm30_stop_capture (void)
  映像のフレームバッファへのキャプチャを停止する
  戻り値　：0 正常終了、1 エラー

- int nm30_disconnect (void)
  カメラへの接続を切断する
  戻り値　：0 正常終了、1 エラー

8. サンプルプログラム

- サンプル 1

このサンプルプログラムは、映像フレームを 1 枚取得し、JPEG データをファイルに保存します。

CaptureSingleFrame()　を実行すると、

接続カメラ数のチェック、カメラ 1 の選択、初期化
→映像フレーム取得→JPEG データをファイルへ保存
→カメラ切断

の順に処理を行い、終了します。

サンプル1

```
#include "SSK.h"

// プロトタイプ宣言
int         CaptureSingleFrame( void );
int         SaveImage( BYTE *p, long len );

// 関数名：CaptureSingleFrame()
// 機能　：
//    カメラから画像を枚取り込み、
//    ファイルへ保存します
//
//    処理手順
//      カメラ接続
//      →画像フレーム枚取得→JPEGファイル保存
//      →カメラ切断
//
// 戻り値：(正常終了) 0以外(処理異常)
//
int CaptureSingleFrame( void )
{
        int ret;
        int fret;       // 本関数戻り値
        ImageStruct *imageInfo;         // 映像フレーム情報

        fret = 0;
        imageInfo = NULL;

        // 接続カメラ数のチェック
        ret = nm30_get_number();
        if ( ret <= 0 ) return -1;

        // カメラ 1 を選択
        ret = nm30_select(1);
        if ( ret != 0 ) return -2;

        // カメラ初期化
        ret = nm30_init();
        if ( ret != 0 ) return -3;

        // フレームバッファ(DLL内部)へのキャプチャ開始
        ret = nm30_start_capture();
        if ( ret != 0 ) {
            fret = -4;
```

```
        goto CAM_CLOSE;
    }

    // キャプチャサイズ設定
    ret = nm30_set_capture_size( 640, 480 );        // VGA
    if ( ret != 0 ) {
        fret = -5;
        goto CAM_CLOSE;
    }

    // 映像フレームをキャプチャ(この時点の画像がキャプチャされる)
    ret = nm30_grab_frame(0);
    if ( ret != 1 ) {
        fret = -6;
        goto CAM_CLOSE;
    }

    // 画像データへのポインタを取得
    imageInfo = nm30_retrieve_frame();
    if ( imageInfo == NULL ) {
        fret = -7;
        goto CAM_CLOSE;
    }

    // JPEGファイルへ保存
    SaveImage( imageInfo->StreamImageBytes, imageInfo->StreamLength );

CAM_CLOSE:
    // フレームバッファへのキャプチャの停止
    ret = nm30_stop_capture();
    if ( ret != 0 ) fret = -8;

    // カメラ切断
    ret = nm30_disconnect();
    if ( ret != 0 ) fret = -9;

    return fret;

}

// 画像データをファイルへ保存
int     SaveImage( BYTE *p, long len )
{
    HANDLE hf;

    hf = CreateFile( L"NM30Image.jpg", GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,
        FILE_ATTRIBUTE_NORMAL, NULL );
    if ( hf == NULL ) {
        return -1;
    }

    DWORD wlen;
    BOOL bret;
    bret = WriteFile( hf, p, len, &wlen, NULL );

    CloseHandle(hf);

    if ( bret == false ) return -2;
    return 0;
}
```

(DLL 監修　産業技術総合研究所 殿)