

SVO-03  
画像出力ライブラリ説明書

Rev.1.30

株式会社ネットビジョン

## 改定履歴

版数	日付	内容	担当	備考
1.00	2015/03/20	・新規作成	柏木	
1.10	2017/03/08	・SVO-03 ボード FW200,FPGA98 用に変更	柏木	
1.20	2017/08/18	・SVOUSB30_OfflineOutputStartEx API を追加 ・SVOUSB30_GetOutputStatusEx API の説明を追加	柏木	
1.30	2017/08/24	・ヘッダーの Rev 表記を修正 ・改定履歴にて 1.10 版の日付を修正 ・「3.2. SVO-03 画像出力ライブラリ仕様」を修正 ・「3.3.12. SVOUSB30_OfflineOutputStartEx」にて誤記を修正 ・「3.3.19. PC から随時画像データを転送する長時間出力時の画像出力ライブラリ使用例」にて誤記を修正	柏木	

## 目次

1. 適用 .....	3
2. 概要 .....	3
3. 仕様 .....	3
3.1. ファイル構成 .....	3
3.2. SV0-03画像出力ライブラリ仕様 .....	4
3.3. SV0-03画像出力ライブラリAPIリファレンス .....	5
3.3.1. SVOUSB30_Init .....	5
3.3.2. SVOUSB30_EnumDevice .....	5
3.3.3. SVOUSB30_End .....	5
3.3.4. SVOUSB30_Open .....	6
3.3.5. SVOUSB30_Close .....	6
3.3.6. SVOUSB30_ReadReg .....	7
3.3.7. SVOUSB30_WriteReg .....	7
3.3.8. SVOUSB30_GetBoardStatus .....	8
3.3.9. SVOUSB30_SetParam3 .....	9
3.3.10. SVOUSB30_OfflinePictureSendEx .....	11
3.3.11. SVOUSB30_OfflineOutputStart .....	13
3.3.12. SVOUSB30_OfflineOutputStartEx .....	14
3.3.13. SVOUSB30_OfflineOutputStop .....	15
3.3.14. SVOUSB30_GetOutputStatusEx API .....	16
3.3.15. SVOUSB30_GpioAcc .....	17
3.3.16. SVOUSB30_GetVersion .....	17
3.3.17. エラーコード一覧 .....	18
3.3.18. SDRAM内画像繰り返し出力時の画像出力ライブラリ使用例 .....	19
3.3.19. PCから随時画像データを転送する長時間出力時の画像出力ライブラリ使用例 .....	21
3.3.20. PCから随時画像データを転送する長時間出力時の画像データ送信について .....	24
3.4. 画像出力ライブラリーを使用したサンプルプログラムの提供 .....	24

## 1. 適用

本説明書は USB3.0 画像出力ボード SVO-03 システムに適用します。

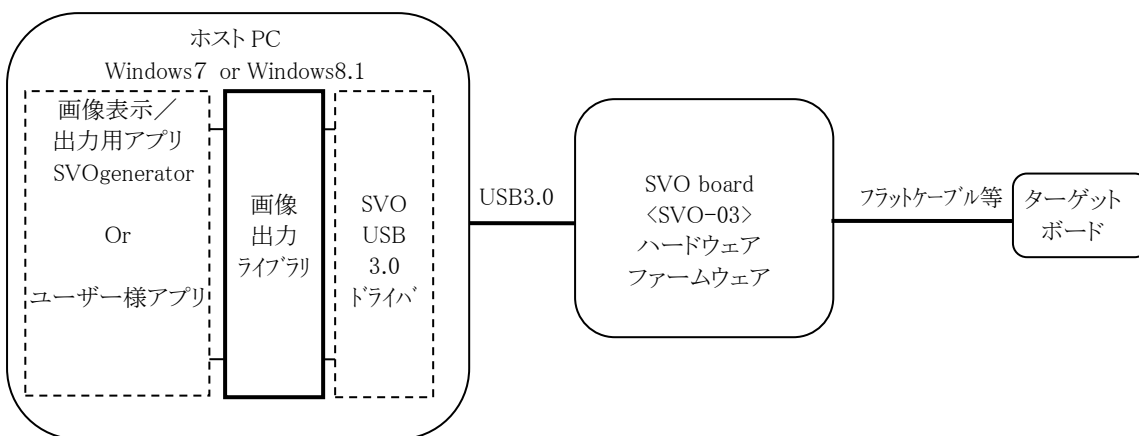
## 2. 概要

SVO とは画像データなどのデジタルデータを出力するイメージ・ジェネレータで Windows 上のソフトウェアとハードウェア及びファームウェアから構成されます。

本説明書では SVO-03 ボードを USB3.0 を介して接続、制御するための画像出力ライブラリについて記述します。

画像出力ライブラリを使用することで、ホスト PC でカメラから取り込んだ画像データまたは作成された画像データを SVO-03 ボードから指定されたビデオタイミングで出力することができます。

【図1】システム構成図



## 3. 仕様

本ライブラリは SVO 用 USB3.0 デバイスドライバを呼び出し SVO-03 ボードより画像を出力するためのライブラリです。アプリケーションからは本ライブラリを使用して制御します。本ライブラリ内では CreateFile API でドライバをオープンし、ReadFile、WriteFile、DeviceIoControl API をコールすることにより SVO-03 ボードの制御及び画像データの送信を実現します。SVO-03 ボードに対するパラメータの設定、ステータスの取得などは DeviceIoControl API を使用します。画像データ、ファームウェア、FPGA データのアップデート用のデータの送信には WriteFile API を使用します。

### 3.1. ファイル構成

本ライブラリは以下のファイルを提供します。

- ・SVO 用 USB3.0 デバイスドライバ  
Software CD-ROM の Driver\_x86、Driver\_x64 フォルダに格納
- ・SVOUSB30.H (Software CD-ROM の“Src¥SVO 画像出力ライブラリ”フォルダに格納)  
本ライブラリを使用する際に必要なインクルードファイルです。
- ・SVOUSB30.DLL (Software CD-ROM の“SVO-AP\_x86”、“SVO-AP\_x64”フォルダに格納)  
本ライブラリです。
- ・SVOUSB30.LIB (Software CD-ROM の“画像出力ライブラリ”フォルダに格納)  
本ライブラリリンクモジュールです。

## 3.2. SVO-03 画像出力ライブラリ仕様

アプリケーションからは下表のAPIを介して SVO-03 ボードを制御します。

API名	機能	備考
SVOUSB30_Init	本ライブラリを初期化します	
SVOUSB30_EnumDevice	SVO-03 ボードの接続数を確認します	
SVOUSB30_End	本ライブラリを終了します	
SVOUSB30_Open	SVO 用 USB3.0 デバイスドライバをオープンします	番号指定
SVOUSB30_Close	SVO 用 USB3.0 デバイスをクローズします	番号指定
SVOUSB30_ReadReg	SVO-03 ボード・レジスタから値を読み込みます	番号指定
SVOUSB30_WriteReg	SVO-03 ボード・レジスタへ値を書き込みます	番号指定
SVOUSB30_GetBoardStatus	SVO-03 ボードのボード情報を取得します	番号指定
SVOUSB30_SetParam3	SVO-03 ボードへ出力設定情報を通知します	番号指定
SVOUSB30_OfflinePictureSendEx	SVO-03 ボードへ画像データを送信します	番号指定
SVOUSB30_OfflineOutputStart	SVO-03 ボードに対してオフライン画像出力を開始します	番号指定
SVOUSB30_OfflineOutputStartEx	SVO-03 ボードに対して出力モードを指定してオフライン画像出力を開始します	番号指定
SVOUSB30_OfflineOutputStop	SVO-03 ボードに対してオフライン画像出力を停止します	番号指定
SVOUSB30_GetOutputStatusEx	SVO-03 ボードの画像出力時の情報を取得します	番号指定
SVOUSB30_GpioAcc	SVO-03 ボード上の GPIO ポートを制御します	番号指定
SVOUSB30_GetVersion	SVOUSB30.dll のバージョン番号を取得します。	

※番号指定とは、SVO-03 ボードの**ボード番号**を**引数に指定することが必須であることを示します**。SVO-03 ボードのボード番号は SVOUSB30\_EnumDevice API で取得することができます。

※SVOUSB30.H には上記に含まれない API も定義していますが、未実装 API 及びユーザー使用不可 API になりますので使用しないでください。

※SVO-02 用画像出力ライブラリ向けで作成されたプログラムは使用できません。

### 3.3. SVO-03 画像出力ライブラリ API リファレンス

#### 3.3.1. SVOUSB30\_Init

API SVOUSB30\_Init  
機能 本ライブラリの内部変数を初期化します

プロトタイプ  
void SVOUSB30\_Init( void );

戻り値  
なし

備考  
・必ず最初に呼び出して下さい。

#### 3.3.2. SVOUSB30\_EnumDevice

API SVOUSB30\_EnumDevice  
機能 SVO-03 ボードの接続台数及び接続された SVO-03 ボードの番号を接続台数分取得します

プロトタイプ  
DWORD SVOUSB30\_EnumDevice (   
    ULONG \*pulSVO\_Num, // 接続台数を格納するポインタ  
    ULONG \*pulSVO\_NumTable // 接続台数分のボード番号を格納する配列のポインタ  
// このポインタが NULL の場合、接続台数のみ格納します  
);

戻り値  
SVOUSB30\_RET\_NORMAL 正常終了  
SVOUSB30\_RET\_ERROR\_DEVOPEN SVO-03 ボードが接続されていないか、SVO-03 ボードの電源が入っていません  
SVOUSB30\_RET\_ERROR\_PARAMETER 引数に間違いがあります  
その他 Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

備考  
・必ず SVOUSB30\_Init API の後に呼び出して下さい。  
・本 API は最初に接続台数を検知するために呼び出し、接続台数分の配列(メモリー)を確保した後、再度呼び出しで各 SVO-03 ボードのボード番号を取得できます。

#### 3.3.3. SVOUSB30\_End

API SVOUSB30\_End  
機能 本ライブラリの終了処理を行います

プロトタイプ  
void SVOUSB30\_End( void );

戻り値  
なし

備考  
・必ず最後に呼び出して下さい。

## 3.3.4. SVOUSB30\_Open

API SVOUSB30\_Open

機能 指定の SVO-03 ボード用の SVO ドライバをオープンします

プロトタイプ

```

DWORD SVOUSB30_Open (
    ULONG        ulBoardNum        // SVO-03 ボード番号を指定します
);

```

戻り値

SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_ERROR_DEVOPEN	SVO-03 ボードが接続されていないか、SVO-03 ボードの電源が入っていません
SVOUSB30_RET_ERROR_MULTIOPEN	同じアプリケーションからは 2 重にオープンできません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

備考

・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。

## 3.3.5. SVOUSB30\_Close

API SVOUSB30\_Close

機能 指定の SVO-03 ボード用の SVO ドライバをクローズします

プロトタイプ

```

DWORD SVOUSB30_Close (
    ULONG        ulBoardNum        // SVO-03 ボード番号を指定します
);

```

戻り値

SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります

備考

・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。

## 3.3.6. SVOUSB30\_ReadReg

API SVOUSB30\_ReadReg

機能 指定の SVO-03 ボードの SVO-03 ボード・レジスタから値を読み込みます

## プロトタイプ

```

DWORD SVOUSB30_ReadReg (
    ULONG    ulBoardNum    // SVO-03 ボード番号を指定します
    ULONG    ulAddr,       // SVO-03 ボード・レジスタ・アドレス
    ULONG    ulDataWicth,  // データバイト幅(4 固定)
    PULONG   pulReadData   // 読み込んだ値を格納するポインタ
);

```

## 戻り値

SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

## 備考

- ・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ・SVO-03 ボード・レジスタ・アドレスは「SVO-03 ハードウェア・レジスタ表」をご覧ください。

## 3.3.7. SVOUSB30\_WriteReg

API SVOUSB30\_WriteReg

機能 指定の SVO-03 ボードの SVO-03 ボード・レジスタへ値を書き込みます

## プロトタイプ

```

DWORD SVOUSB30_WriteReg (
    ULONG    ulBoardNum    // SVO-03 ボード番号を指定します
    ULONG    ulAddr,       // SVO-03 ボード・レジスタ・アドレス
    ULONG    ulDataWicth,  // データバイト幅(4 固定)
    ULONG    ulWriteData   // 書き込み値
);

```

## 戻り値

SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

## 備考

- ・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ・SVO-03 ボード・レジスタ・アドレスは「SVO-03 ハードウェア・レジスタ表」をご覧ください。レジスタ表に記載されている範囲外の値やアドレスを書き込んだ場合の動作は保証できません。



## 3.3.8. SVOUSB30\_GetBoardStatus

API SVOUSB30\_GetBoardStatus

機能 SVO-03 ボードのステータス情報(バージョン、ディップスイッチ、メモリサイズなど)を取得します

## プロトタイプ

```
DWORD SVOUSB30_GetBoardStatus (
    ULONG        ulBoardNum,    // SVO-03 ボード番号を指定します
    PGET_BSTATUS pStatus       // ボード・ステータス情報構造体のポインタ
);
```

## 戻り値

SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

## 構造体

```
typedef _GET_BSTATUS {
    ULONG    ulHWVersion;        // FPGA のバージョン番号
    ULONG    ulFWVersion;        // ファームウェアのバージョン番号
    ULONG    ulOPStatus;         // 未使用
    ULONG    ulHWEdition;        // FPGA のエディション番号
    ULONG    ulH_Period;         // 未使用
    ULONG    ulH_SyncStart;       // 未使用
    ULONG    ulH_SyncEnd;        // 未使用
    ULONG    ulDIP_SW2;          // DIP スイッチ S2 モニター
    ULONG    ulV_Period;         // 未使用
    ULONG    ulV_SyncStart;       // 未使用
    ULONG    ulV_SyncEnd;        // 未使用
    ULONG    ulSyncInfo;         // 未使用
    ULONG    ulSyncMonitor;      // 未使用
    ULONG    ulSdramInfo;        // SDRAM 容量(128 or 256 単位 MB)
    ULONG    ulBoardVer;         // SVO ボードバージョン番号 (SVO-02:02h、SVO-03:03h)
    ULONG    ulPIIClock;         // 未使用
} GET_BSTATUS, *PGET_BSTATUS;
```

## 備考

・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。

## 3.3.9. SVOUSB30\_SetParam3

API SVOUSB30\_SetParam3

機能 SVO-03 ボードへ出力設定情報を通知します

プロトタイプ

```
DWORD SVOUSB30_SetParam3 (
    ULONG          ulBoardNum,    // SVO-03 ボード番号を指定します
    PSET_PARAM3    pParam        // 出力設定情報構造体のポインタ
);
```

戻り値

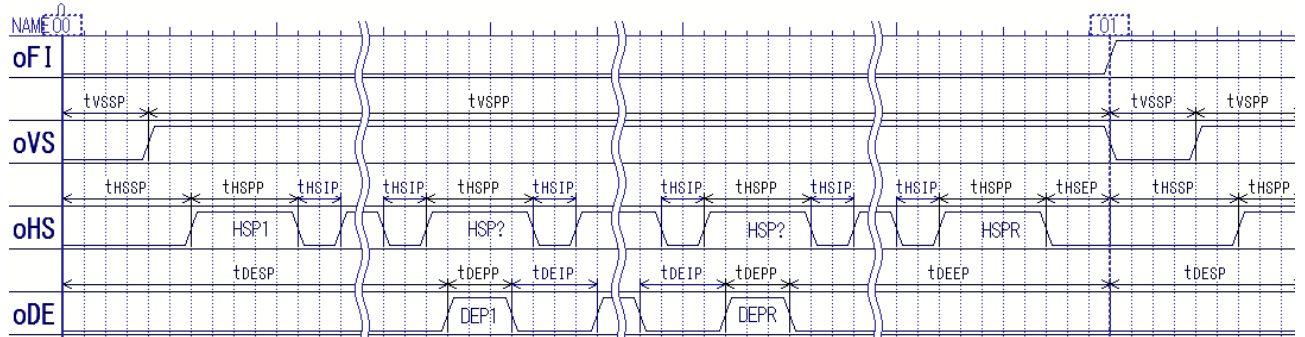
SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

構造体

```
typedef struct _SET_PARAM3 {
    ULONG    ulVSSP_CNT_F1;    // VS 信号の Start Period(スタート期間)クロック数 (Field1)
    ULONG    ulVSSP_CNT_F2;    // VS 信号の Start Period(スタート期間)クロック数 (Field2)
    ULONG    ulVSP_CNT;        // VS 信号の Pulse Period(パルス期間)クロック数
    ULONG    ulHSSP_CNT;       // HS 信号の Start Period(スタート期間)クロック数
    ULONG    ulHSIP_CNT;       // HS 信号の Intermediate Period(中間期間)クロック数
    ULONG    ulHSPP_CNT;       // HS 信号の Pulse Period(パルス期間)クロック数
    ULONG    ulHSPR_CNT_F1;    // HS 信号の Puse Repeat (1 フレーム当たりの繰り返し数) (Field1)
    ULONG    ulHSPR_CNT_F2;    // HS 信号の Puse Repeat (1 フレーム当たりの繰り返し数) (Field2)
    ULONG    ulDESP_CNT;       // DE 信号の Start Period(スタート期間)クロック数
    ULONG    ulDEIP_CNT;       // DE 信号の Intermediate Period(中間期間)クロック数
    ULONG    ulDEPP_CNT;       // DE 信号の Pulse Period(パルス期間)クロック数
    ULONG    ulDEPR_CNT;       // DE 信号の Puse Repeat (1 フレーム当たりの繰り返し数)
    ULONG    ulExternalTrigger; // 外部トリガー設定 (0:OFF, 1:ON)
    ULONG    ulDelayMerginClock; // 外部トリガー時の Delay Clock 数、F-Sync Mergin Clock 数を設定
                                // bit31-16 : DelayClock 数
                                // bit15-00 : F-Sync Mergin 数
    ULONG    ulSyncInfo;       // 同期信号設定情報
                                // bit31-23:予約
                                // bit22-20:SDR_DDR モード
                                // (0:SDR-Pos,1:SDR-Neg,4:DDR 0,5:DDR 180,6:DDR -90,7:DDR 90)
                                // bit19-18:予約
                                // bit17:8bit 出力時のバイトアライメント設定(0:D0-D7 出力,1:D8-D15 出力)
                                // bit16:HSYNC 極性 (0 で LowActive, 1 で HighActive)
                                // bit15-11:bit/clock (0:32,8:8,16:16,24:24)
                                // bit10-07:pictype (未使用)
                                // bit06-05:ByteSwap (未使用)
                                // bit04:1 固定
                                // bit03-02:OneColorByte (0:32,1:8,2:16,3:24)
                                // bit01:DCK出力に関して、ターゲットから見たデータおよび同期信号の
                                // 取込みエッジの極性を指定します。
                                // SDR時0で立ち上りエッジ、1で立ち下りエッジ
                                // DDR 時 0 で先行データを立ち上りエッジ、後続データを立ち下りエッジ
                                // DDR 時 1 で先行データを立ち下りエッジ、後続データを立ち上りエッジ
                                // bit00:VSYNC 極性 (0 で LowActive, 1 で HighActive)
} SET_PARAM3, *PSET_PARAM3;
```

## 備考

- ・SV0-03 ボード番号は、SV0USB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ・構造体メンバ `ulVSSP_CNT_F1` から `ulDEPR_CNT` については、下図を参考にしてください。



## 3.3.10. SVOUSB30\_OfflinePictureSendEx

API SVOUSB30\_OfflinePictureSendEx  
機能 SVO-03 ボードへ画像データを送信します(ボードへの指示フラグ付き)

## プロトタイプ

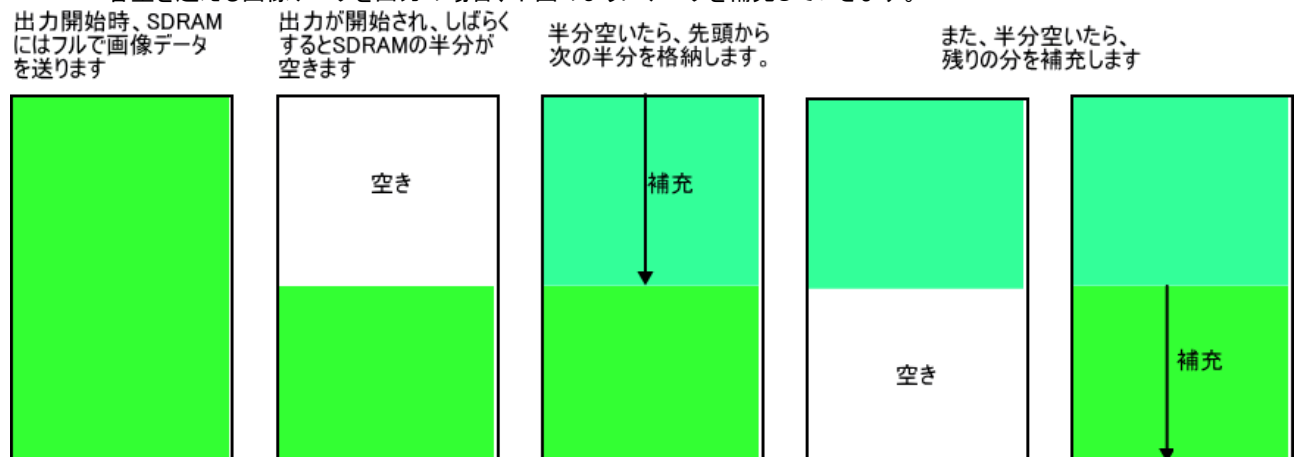
```
DWORD SVOUSB30_OfflinePictureSendEx (
    ULONG    ulBoardNum,    // SVO-03 ボード番号を指定します
    LPVOID    lpFrameBuf,    // 画像データ格納バッファのポインタ
    ULONG    ulLen,        // 画像データのバイト数
    ULONG    ulFrameNum,    // 画像データ内のフレーム数
    ULONG    ulAddr,        // 画像データ格納先アドレス
    ULONG    ulFlag,        // ボードへの指示フラグ
                        // 0 = 最初の転送
                        // 1 = SDRAM 前半部分への転送
                        // 2 = SDRAM 後半部分への転送
    PULONG    pulRetLen      // 実際に送信したバイト数を格納するポインタ
);
```

## 戻り値

SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

## 備考

- ・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ・送信できる画像データの最大バイト数(ulLen)は SVOUSB30\_GetBoardStatus API で取得した SDRAM 容量を超えないように注意してください。
- ・画像データのバイト数(ulLen)は 64 で割り切れるようにしてください。
- ・画像データ内の 1 フレームのバイト数も 64 で割り切れるようにしてください。1 フレームのバイト数が 64 で割り切れると、画像データのバイト数も 64 で割り切れることになります。
- ・画像データ格納先アドレスは SVOUSB30\_GetBoardStatus API で取得した SDRAM 容量および画像データのバイト数を考慮して指定してください。
- ・SDRAM 容量を超える画像データを出力の場合、SVOUSB30\_GetOutputStatusEx API、SVOUSB30\_OfflinePictureSendEx API で出力を制御します。
- ・SDRAM 容量を超える画像データを出力の場合、下図のようにデータを補充していきます。



・画像データ 1 フレームは下図のように計算、詰めて格納してください。

1 フレーム
$1,600 \times 1,200 \times 2 = 3,840,000$ バイト
2 フレーム
$1,600 \times 1,200 \times 2 = 3,840,000$ バイト
3 フレーム
$1,600 \times 1,200 \times 2 = 3,840,000$ バイト
4 フレーム
$1,600 \times 1,200 \times 2 = 3,840,000$ バイト
$3,840,000 \times 4 = 15,360,000$ バイト

1,600x1,200 の RAW-10 ビットデータの場合、1 画素には 16 ビット必要で、1 フレームのバイト数を計算すると、

$1,600 \times 1,200 \times 2 = 3,840,000$  バイトになります。3,840,000 は 64 で割り切れるので、パディングは必要ありません。

左図では、4 フレームあるので、画像データのバイト数は

$3,840,000 \times 4 = 15,360,000$  バイトです。

1 フレームバイト数が 64 で割り切れない場合は、1 フレーム単位で 64 で割り切れるようパディングが必要です。

## 3.3.11. SVOUSB30\_OfflineOutputStart

API SVOUSB30\_OfflineOutputStart

機能 SVO-03 ボードに対してオフライン画像出力を開始します

## プロトタイプ

```

DWORD SVOUSB30_OfflineOutputStart (
    ULONG    ulBoardNum,        // SVO-03 ボード番号を指定します
    ULONG    ulStartFrameNum,   // 出力を開始するフレーム番号 (0~9999)
    ULONG    ulEndFrameNum,     // 出力を終了するフレーム番号 (0~9999)
    ULONG    ulCycleCount       // 繰り返しフラグ
                                // 0:SDRAM 内画像出力で繰り返し無し
                                // 1:SDRAM 内画像出力で繰り返し有り
                                // 2:PC から随時画像出力で繰り返し無し
                                // 3:PC から随時画像出力で繰り返し有り
);

```

## 戻り値

SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_NORMAL_OUT	正常終了(出力中)
SVOUSB30_RET_NORMAL_STOP	正常終了(停止中)
SVOUSB30_RET_ERROR_NODATA	出力する画像データが格納されていません
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
SVOUSB30_RET_ERROR_SEQ_INVALID	シーケンスが不正です(出力情報が設定されていません)
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

## 備考

- ・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ・ulCycleCount(繰り返し回数)が 1 の場合は、SVOUSB30\_OfflineOutputStop API が実行されるまで、無限に繰り返します。
- ・本 API を発行する前に、SVOUSB30\_OfflinePictureSendEx API にて画像データを SVO-03 ボードに送信、格納する必要があります。

## 3.3.12. SVOUSB30\_OfflineOutputStartEx

API SVOUSB30\_OfflineOutputStartEx

機能 SVO-03 ボードに対して出力モードを指定して画像出力を開始します

## プロトタイプ

```

DWORD SVOUSB30_OfflineOutputStart (
    ULONG    ulBoardNum,        // SVO-03 ボード番号を指定します
    ULONG    ulOpeMode         // 出力モード ※備考参照
                                // 0 : Normal
                                // 1 : Repeat
                                // 2 : Reentering
);

```

## 戻り値

SVOUSB30_RET_NORMAL	正常終了
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
SVOUSB30_RET_ERROR_SEQ_INVALID	シーケンスが不正です(出力情報が設定されていません)
その他	Win32API エラー (bit31-28:EH, bit27-0:GetLastError 戻り値)

## 備考

- ・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ・ulOpeMode は画像データがメモリーにない時の振る舞いを指定します。1 フレーム転送を完了した後、次のフレーム転送を行う際に直ちに読出し可能でない場合にどうするかを指定します。

// 0 : Normal	次のフレーム読出しが可能でない場合に待機する。ターゲットに対し、ホスト PC からフレームを常に転送しながら出力する場合で、間に合わない場合にブランク期間を延長しても良い場合、または一定のタイミングの保証よりも同じフレームがダブらないことを優先する場合に使用する。
// 1 : Repeat	1 フレーム出力後、次のフレームが読出し可能でない場合、直前に出力したフレームを再出力する。ターゲットへの常に一定のタイミングでのフレーム出力を保証しなければならない場合に使用する。
// 2 : Reentering	1 フレーム出力後、読みだしたフレームの内容が、再び <b>FrameMemory</b> に書き込まれたものとしてフレーム管理テーブルを更新する。FrameMemory に複数のフレームを予め書き込んでおき、本モードを使用することで、FrameMemory に格納済みのフレームを決まった順序とタイミングで、繰り返し出力する場合に使用する。

- ・画像データの送信は別途 SVOUSB30\_OfflinePictureSendEx API にて行います。

## 3.3.13. SVOUSB30\_OfflineOutputStop

API SVOUSB30\_OfflineOutputStop

機能 SVO-03 ボードに対してオフライン画像出力を停止します

プロトタイプ

```
DWORD SVOUSB30_OfflineOutputStop (
    ULONG    ulBoardNum,        // SVO-03 ボード番号を指定します
);
```

戻り値

SVOUSB30\_RET\_NORMAL 正常終了

SVOUSB30\_RET\_ERROR\_NOOPEN オープンされていません

その他 Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

備考

・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。

・本 API を発行する前に、SVOUSB30\_OfflineOutputStart API にてオフライン出力を開始する必要があります。



## 3.3.14. SVOUSB30\_GetOutputStatusEx API

API SVOUSB30\_GetOutputStatusEx  
機能 画像出力時の情報を取得します

## プロトタイプ

```
DWORD SVOUSB30_GetOutputStatusEx (
    ULONG    ulBoardNum,    // SVO-03 ボード番号を指定します
    PULONG    pulFrameNum,  // DLB ステータスを格納するポインタ
    PULONG    pulTotalNum,  // DLB フレームカウンタを格納するポインタ
    PULONG    pulFreeMem    // DLB フレーム R/W カウンタを格納するポインタ
);
```

## 戻り値

SVOUSB30_RET_NORMAL	正常終了
<del>SVOUSB30_RET_NORMAL_OUT</del>	<del>正常終了(出力中)</del>
<del>SVOUSB30_RET_NORMAL_STOP</del>	<del>正常終了(停止中)</del>
SVOUSB30_RET_ERROR_NOOPEN	オープンされていません
SVOUSB30_RET_ERROR_PARAMETER	引数に間違いがあります
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

## 備考

・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。

・本 API は長時間出力時に使用してください。

・本 API の戻り値により、出力状態を確認することが可能です。

・pulFrameNum にて、フレームに空きがあるか確認できます。

Bit23 FACTFLL	フレームメモリがフル
Bit16 FACTEMP	フレームメモリが空
Bit15 FWFL	Write 側から見てフレームメモリがフル
Bit12 FWRDY	Write 側から見てフレームメモリが書き込み可能
Bit8 FWEMP	Write 側から見てフレームメモリが空
Bit7 FRFLL	Read 側から見てフレームメモリがフル
Bit4 FRRDY	Read 側から見てフレームメモリが読み込み可能
Bit0 FREMP	Read 側から見てフレームメモリが空
その他無効	

・pulTotalNum にて、フレーム出力ステータスが確認できます。

Bit31-24 FACTMAX	フレームメモリに格納する最大フレーム数(予め設定しておきます)
Bit23-16 FACTCNT	出力可能なフレーム数
Bit15-8 FWCNT	Write 側から見て書き込み済みフレーム数
Bit7-0 FRCNT	Read 側から見て読み込み可能フレーム数

・pulFreeMem にて、フレーム出力時のライトポインタ、リードポインタが確認できます。

Bit15-8 FWPTR	Write 側から見た書き込みポインタ
Bit7-0 FRPTR	Read 側から見た読み込みポインタ

## 3.3.15. SVOUSB30\_GpioAcc

API SVOUSB30\_GpioAcc  
 機能 SVO-03 ボード上の GPIO ポートを制御します  
 プロトタイプ  
 DWORD SVOUSB30\_GpioAcc (  
     ULONG ulBoardNum, // SVO-03 ボード番号を指定します  
     ULONG ulMode, // 入出力フラグ (0: 入力、1: 出力)  
     ULONG ulPortNum, // ポート番号 (0~5)  
     PULONG pulData // 入出力データ (0 or 1) を格納するポインタ  
 );  
 戻り値  
 SVOUSB30\_RET\_NORMAL 正常終了  
 SVOUSB30\_RET\_ERROR\_PARAMETER 引数に間違いがあります  
 SVOUSB30\_RET\_ERROR\_NOOPEN オープンされていません  
 その他 Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

## 備考

- ・SVO-03 ボード番号は、SVOUSB30\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ・本 API では 1 回の呼び出しで 1 ポートのみ制御となります。

## 3.3.16. SVOUSB30\_GetVersion

API SVOUSB30\_GetVersion  
 機能 SVOUSB30.DLL のバージョン情報を取得します  
 プロトタイプ  
 DWORD SVOUSB30\_GetVersion(  
     char \*pcVerBuf // バージョン番号文字列を格納するポインタ  
 );  
 戻り値  
 SVOUSB30\_RET\_NORMAL 正常終了  
 SVOUSB30\_RET\_ERROR\_PARAMETER 引数に間違いがあります (ポインタが NULL)

## 備考

- ・以下のように文字列ポインタにバージョン番号が格納されます。

例) バージョン番号が 1. 0. 0. 0 の場合

```

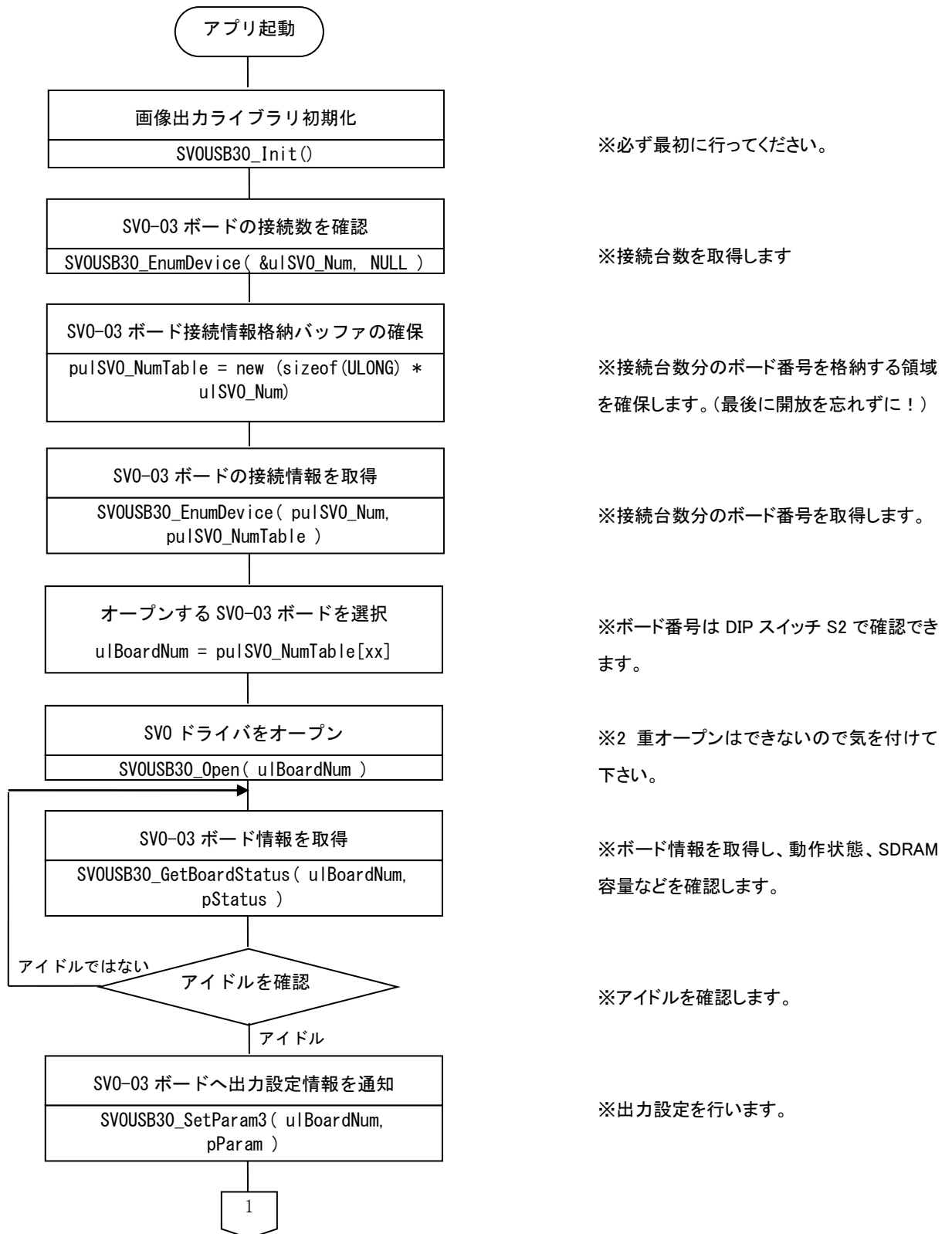
*(pcVerBuf+0) = '1' // 0x31
*(pcVerBuf+1) = '.' // 0x2e
*(pcVerBuf+2) = '0' // 0x30
*(pcVerBuf+3) = '.' // 0x2e
*(pcVerBuf+4) = '0' // 0x30
*(pcVerBuf+5) = '.' // 0x2e
*(pcVerBuf+6) = '0' // 0x30
*(pcVerBuf+7) = '\0' // 0x00

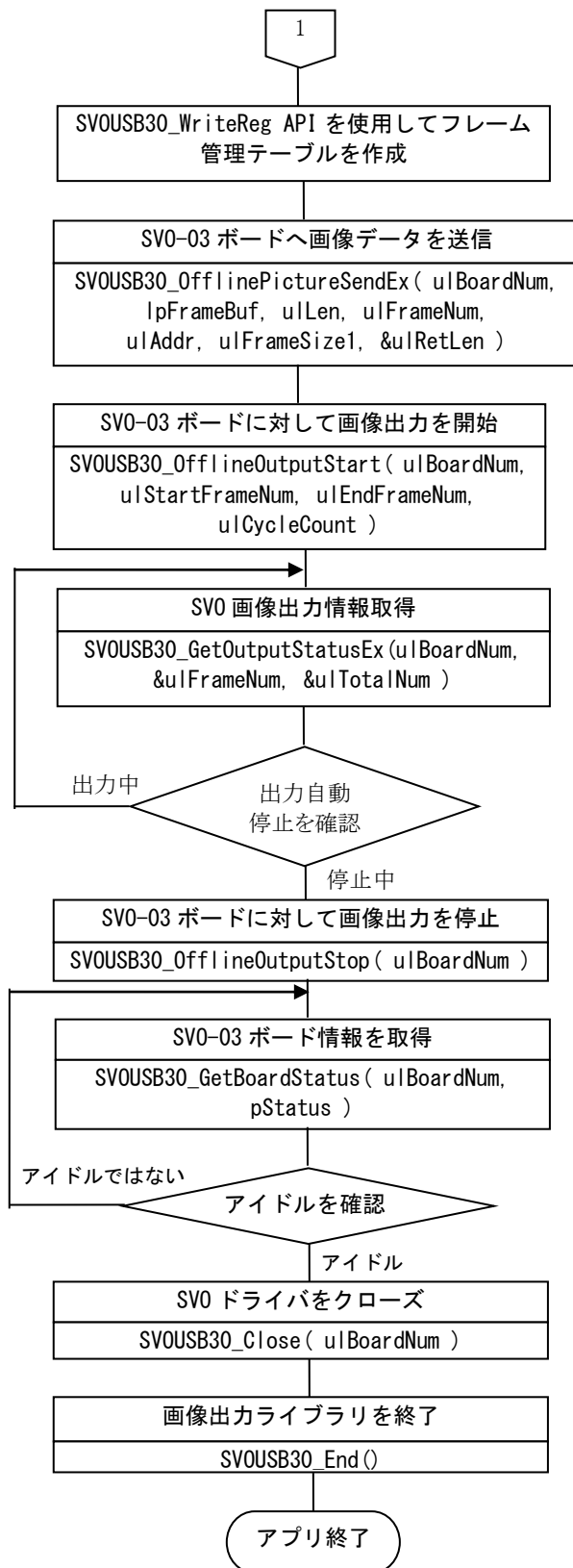
```

## 3.3.17. エラーコード一覧

エラーコード名	コード	内容
SVOUSB30_RET_NORMAL	0x00000001	正常終了
SVOUSB30_RET_NORMAL_OUT	0x00000002	正常終了(出力中)
SVOUSB30_RET_NORMAL_STOP	0x00000003	正常終了(停止中)
SVOUSB30_RET_ERROR_DEVICE	0xe0000000	Win32API エラー発生時に GetLastError の値に OR され
SVOUSB30_RET_ERROR_DEVOPEN	0xf0000001	SVO-03 ボードが接続されていないか、電源が入っていない
SVOUSB30_RET_ERROR_MULTIOPEN	0xf0000002	SVO-03 ボードを 2 重にオープンしようとした
SVOUSB30_RET_ERROR_NOOPEN	0xf0000003	SVO-03 ボードがオープンされていない
SVOUSB30_RET_ERROR_PARAMETER	0xf0000004	パラメーターエラー発生
SVOUSB30_RET_ERROR_NOTIDLE	0xf0000005	SVO-03 ボードがアイドル状態ではない
SVOUSB30_RET_ERROR_FMUPDATE_T	0xf0000006	ファームウェア更新タイムアウトエラー発生
SVOUSB30_RET_ERROR_FPUPDATE_T	0xf0000007	FPGA データ更新タイムアウトエラー発生
SVOUSB30_RET_ERROR_BUSY	0xf1000011	ビジーでコマンドが実行できない
SVOUSB30_RET_ERROR_NODATA	0xf1000012	SVO-03 ボードに画像データが格納されていない
SVOUSB30_RET_ERROR_CMD_INVALID	0xf1000021	コマンドが不正である
SVOUSB30_RET_ERROR_PRM_INVALID	0xf1000022	パラメータが不正である
SVOUSB30_RET_ERROR_CMD_NOIMPLE	0xf1000023	SVO-03 ボードでサポートしていないコマンドである
SVOUSB30_RET_ERROR_SEQ_INVALID	0xf1000023	シーケンスが不正である
SVOUSB30_RET_ERROR_UPDATE_INVALID	0xf10000e1	モジュールデータが不正である
SVOUSB30_RET_ERROR_FROMERS	0xf10000e2	フラッシュメモリーの消去に失敗した
SVOUSB30_RET_ERROR_FROMWT	0xf1000043	フラッシュメモリーの書き込みに失敗した
SVOUSB30_RET_ERROR_INTERNAL	0xf10000f1	内部エラーが発生した
SVOUSB30_RET_ERROR_RESOURCE	0xf10000f2	内部リソースが不足して処理が実行できない

## 3.3.18. SDRAM 内画像繰り返し出力時の画像出力ライブラリ使用例





※SDRAM 内のフレームをどのように配置して出力するかを指定します。

※画像データを送信します。画像データバイト数は 64 で割り切れるようにしてください。

※画像出力開始です。

※繰り返し回数が 0 でなければ自動で出力を停止します。0 の場合は SV0USB30\_OfflineOutputStop API で強制停止させる必要があります。

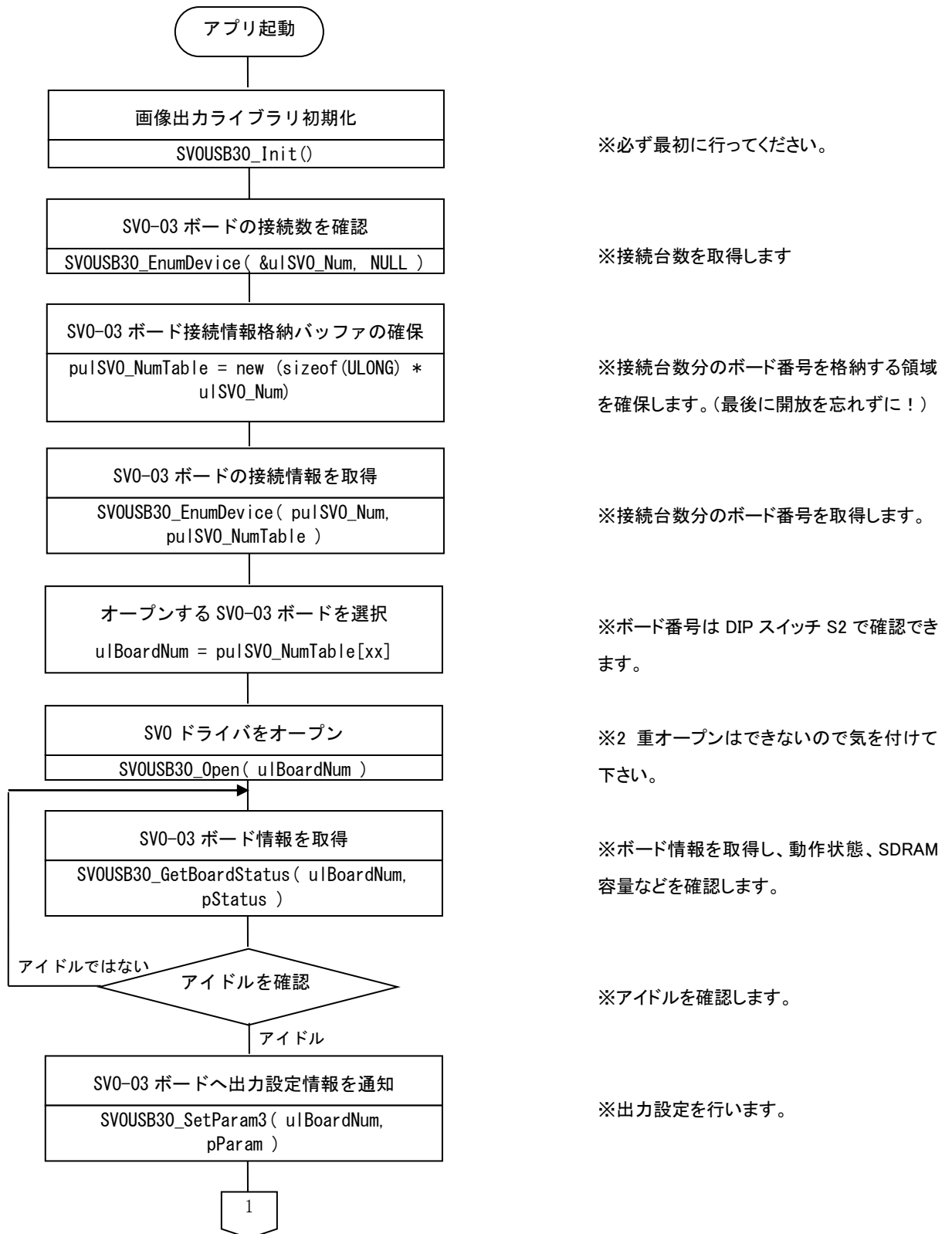
※繰り返し回数が 0 でなければ自動で出力を停止しますが、この API を発行するようにしてください。

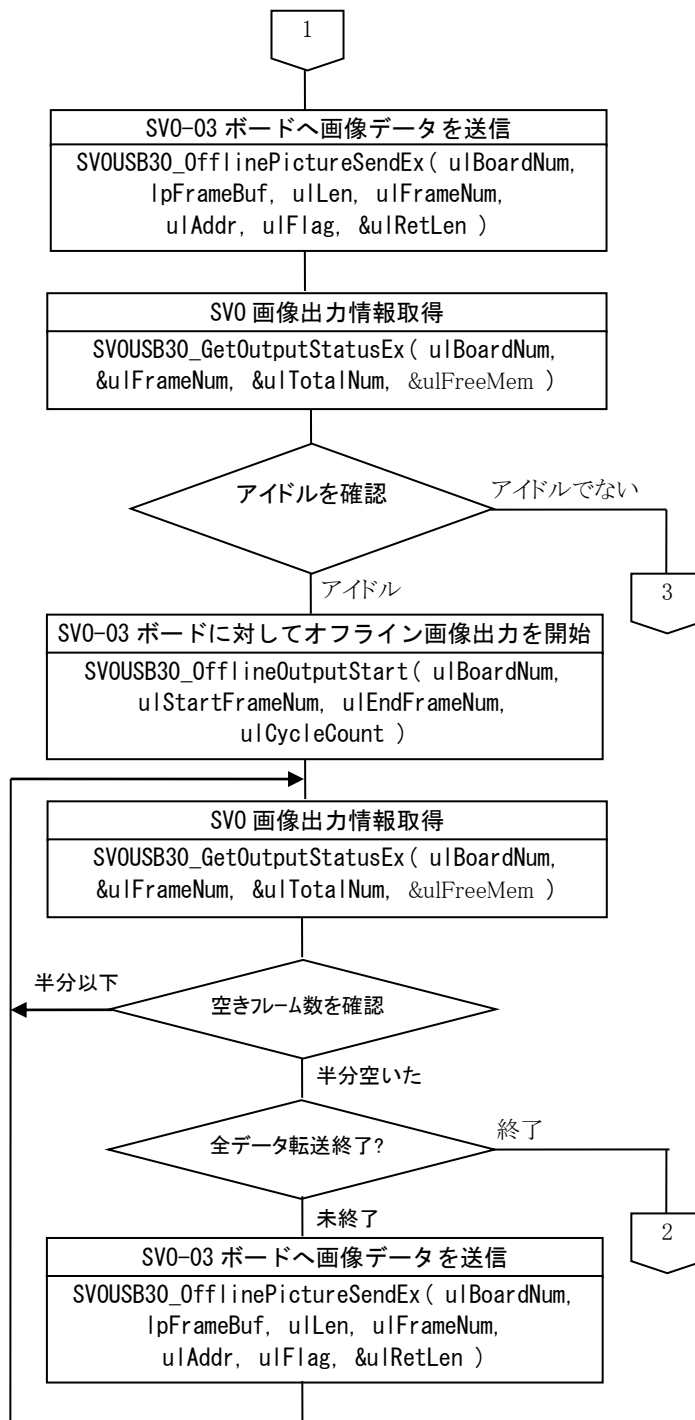
※アイドルになるまで確認します。

※ドライバをクローズします。

※本ライブラリに対して終了を通知します。アプリで確保した領域など開放を行ってください。

## 3.3.19. PC から随時画像データを転送する長時間出力時の画像出力ライブラリ使用例





※最初に SDRAM いっぱいまで画像データを送信します。画像データバイト数は64で割りきれられるようにしてください。ulFlag=1

※出力情報を取得します。

※アイドルを確認します。アイドルでなければ終了します。

※画像出力開始です。

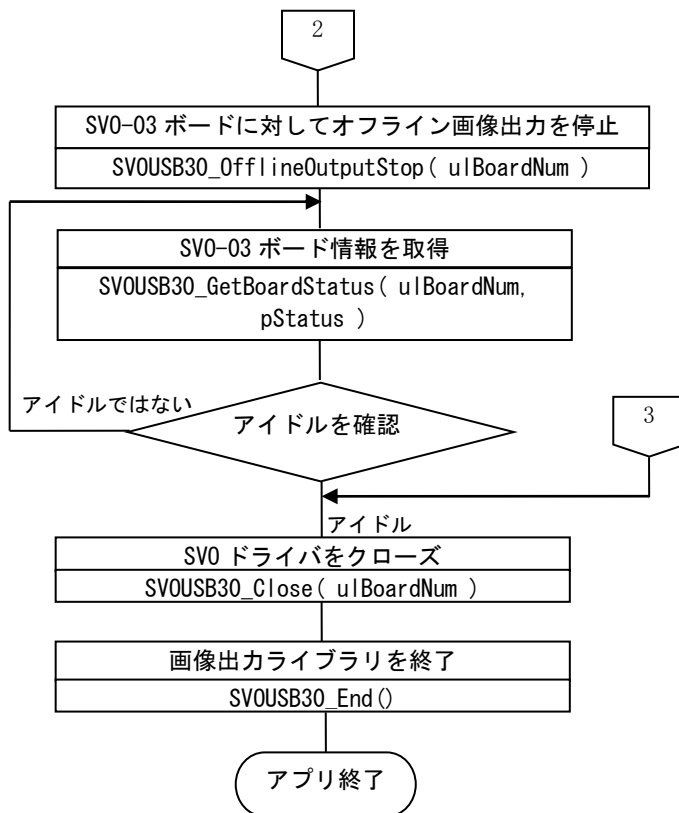
※出力情報を取得し、エラー発生や、SDRAM の空き情報を取得します。

※空きフレーム数が半分以上になるまで確認します。

※全データ転送済みであれば、終了処理を行います。

※SDRAM に空きフレーム数分、画像データを補充します。SDRAM のどこに補充するかを意識して引数を設定してください。

ulFlag=0



※発行するようにしてください。

※アイドルになるまで確認します。

※ドライバをクローズします。

※本ライブラリに対して終了を通知します。  
アプリで確保した領域など開放を行ってください。



### 3.3.20. PC から随時画像データを転送する長時間出力時の画像データ送信について

画像の送信は、まず SVO-03 のメモリ分(SDRAM)のフレームを送信し、その後 SVO-03 ボードの空きメモリを確認し、総フレーム数の半分が送信し終わったら次の送信データをボードへ送ります。

たとえば、SVO-03 ボードの SDRAM 容量が 128MB で 640x480 8bit-YUV の画像の場合は、

$$\text{SDRAM\_SIZE} = 128 \times 1024 \times 1024 = 134217728$$

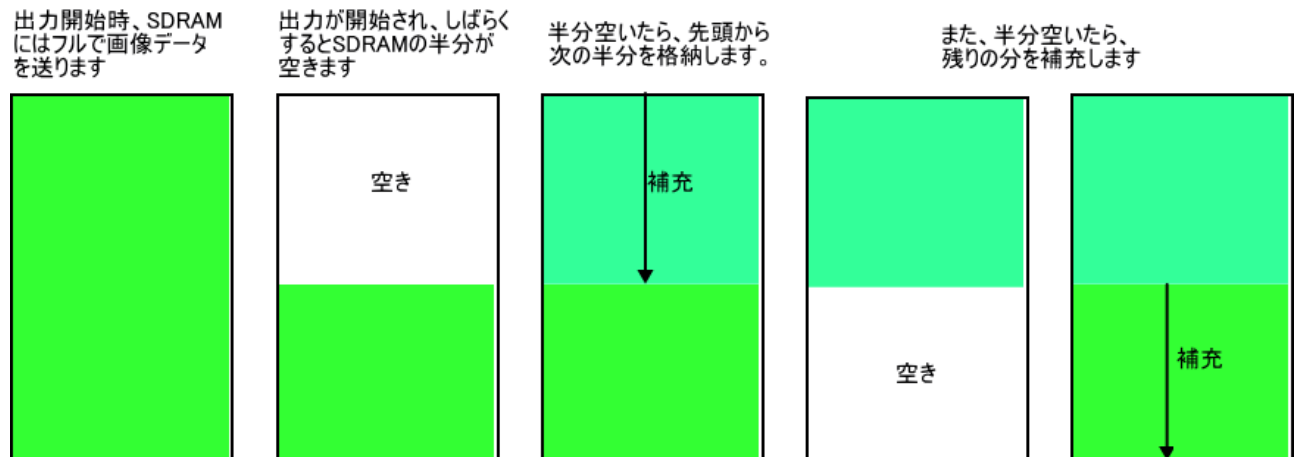
$$\text{1FRAME\_SIZE} = 640 \times 480 \times 2 = 614400$$

$$\text{FRAME\_NUM} = \text{SDRAM\_SIZE} \div \text{1FRAME\_SIZE} \div 2 = 218$$

218 フレームが SDRAM に格納できることになります。インターレースの場合はその倍の 436 フィールド格納できます。

最初に 436 フィールドを SVO-03 ボードへ送信します。SVO-03 ボードから 218 フレーム送信されるとメモリが半分空くので、新たに 218 フレーム送信できることになります。

長時間出力を効率よく行うために、アプリケーションは複数のスレッドを生成し、ファイルから画像データを読み込む処理、画像データを SVO-03 ボードへ転送する処理などマルチスレッドで行うべきです。



### 3.4. 画像出力ライブラリーを使用したサンプルプログラムの提供

2017 年 3 月出荷分から、SVO-03 ボードに付属する Software-CD にはサンプルプログラムを確認していますので参考にしてください。SVO-03 ボードになってから制御が複雑になっており、本説明書を参考にアプリケーションを作成するのは難しいです。