

Sky Vision image Output  
画像出力ライブラリ説明書

Rev.1.07

承認	承認	照査	作成

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
0.01	2006/02/08	新規作成	柏木	
1.00	2006/02/08	誤字脱字を修正しました 「3.3.1. SV0 画像出力ライブラリを初期化します」を変更しました 「3.3.11. SV0 ボードの画像出力情報を取得します」を変更しました 「3.3.12. SV0 ボードへの出力設定情報を通知します」を変更しました 「3.3.15. SV0 ボードへ画像データを送信します」を変更しました	柏木	
1.01	2006/02/09	「3.3.8. USB バス上にバス・リセットを発行します」を変更しました 「3.3.19 エラーコード一覧」を追加しました 「3.3.15. SV0 ボードへ画像データを送信します」を変更しました 「3.3.12. SV0 ボードへの出力設定情報を通知します」に図を追加しました	柏木	
1.02	2006/02/15	「3.3.10. SV0 ボードのボード情報を取得します」を変更しました	柏木	
1.03	2006/02/24	「3.2. SV0 画像出力ライブラリ仕様」の API 一覧を追加しました 「3.3.10. SV0 ボード上の画像格納領域をゼロクリアします」を追加しました	柏木	
1.04	2006/03/02	「3.3.14. SV0 ボードに対してオフライン画像出力を開始します」を変更しました 「3.3.20. エラーコード一覧」を追加しました	柏木	
1.05	2006/03/27	「3.3.20. SVOUSB20. DLL のバージョン情報を取得します」を追加しました	柏木	変更は 赤字参照
1.06	2006/09/27	画像出力ライブラリの各タイトルを機能から関数名に変更しました SVOUSB20_PllUpdate を追加しました SVOUSB20_GetBoardStatus の ulReserve1、2 を以下のように変更しました ● ulReserve1 → ulBoardVer      ボードのバージョン ● ulReserve2 → ulPllClock      PLL Clock 設定値(KHz)	深井	
1.07	2011/09/21	発行社名をスカイウェアからネットビジョンへ変更	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. SVO 画像出力ライブラリ仕様 .....	4
3.3. SVO 画像出力ライブラリ API リファレンス.....	5
3.3.1. SVOUSB20_Init .....	5
3.3.2. SVOUSB20_EnumDevice .....	5
3.3.3. SVOUSB20_End.....	5
3.3.4. SVOUSB20_Open .....	6
3.3.5. SVOUSB20_Close .....	6
3.3.6. SVOUSB20_ReadReg .....	7
3.3.7. SVOUSB20_WriteReg .....	7
3.3.8. SVOUSB20_BusReset .....	8
3.3.9. SVOUSB20_DeviceReset .....	8
3.3.10. SVOUSB20_MemClear .....	8
3.3.11. SVOUSB20_GetBoardStatus .....	9
3.3.12. SVOUSB20_GetOutputStatus.....	11
3.3.13. SVOUSB20_SetParam .....	12
3.3.14. SVOUSB20_OfflineOutputStart.....	14
3.3.15. SVOUSB20_OfflineOutputStop .....	14
3.3.16. SVOUSB20_OfflinePictureSend.....	15
3.3.17. SVOUSB20_FirmwareUpdate.....	16
3.3.18. SVOUSB20_FpgaUpdate .....	16
3.3.19. SVOUSB20_GpioAcc .....	17
3.3.20. SVOUSB20_GetVersion .....	17
3.3.21. SVOUSB20_PllUpdate.....	18
3.3.22. エラーコード一覧.....	19
3.3.23. オフライン出力時の画像出力ライブラリ使用例 .....	20

## 1. 適用

本説明書は Sky Vision image Output（以降 SVO）に適用します。

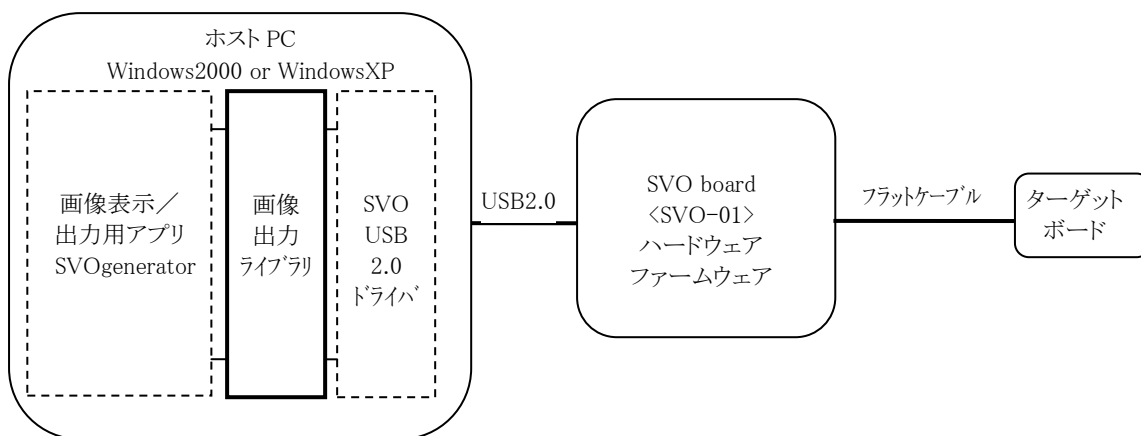
## 2. 概要

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 ハードウェアを USB2.0 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ハードウェア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 SVOUSB2.0 ドライバを呼び出し SVO ハードウェア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 ハードウェアの制御及び画像データの送信を実現します。SVO ハードウェア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。

- ・SVOUSB20.INF（ソフトウェア CD-ROM の DRIVER フォルダに格納）  
SVOUSB2.0 ドライバインストール情報ファイル。
- ・SVOUSB20.SYS（ソフトウェア CD-ROM の DRIVER フォルダに格納）  
SVOUSB2.0 ドライバ本体 Windows ディレクトリ下”System32\drivers”にコピーされ使用されます。
- ・SVOUSB20.H（ソフトウェア CD-ROM の“SOURCE\SVO 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。
- ・SVOUSB20.DLL（ソフトウェア CD-ROM の“SVO-AP”フォルダに格納）  
本ライブラリです。
- ・SVOUSB20.LIB（ソフトウェア CD-ROM の“SOURCE\SVO 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 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 ハードウェアを制御します。

API名	機能	備考
SVOUSB20_Init	本ライブラリを初期化します	
SVOUSB20_EnumDevice	SVO ボードの接続数を確認します	
SVOUSB20_End	本ライブラリを終了します	
SVOUSB20_Open	SVO ドライバをオープンします	番号指定(※)
SVOUSB20_Close	SVO ドライバをクローズします	番号指定
SVOUSB20_ReadReg	SVO ハードウェア・レジスタから値を読み込みます	番号指定
SVOUSB20_WriteReg	SVO ハードウェア・レジスタへ値を書き込みます	番号指定
SVOUSB20_BusReset	USB バス上にバス・リセットを発行します	
SVOUSB20_DeviceReset	SVO ボードを初期化します	番号指定
SVOUSB20_GetBoardStatus	SVO ボードのボード情報を取得します	番号指定
SVOUSB20_GetOutputStatus	SVO ボードの画像出力情報を取得します	番号指定
SVOUSB20_SetParam	SVO ボードへ出力設定情報を通知します	番号指定
SVOUSB20_OfflineOutputStart	SVO 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
SVOUSB20_OfflineOutputStop	SVO 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
SVOUSB20_OfflinePictureSend	SVO ボードへ画像データを送信します	番号指定
SVOUSB20_FirmwareUpdate	SVO ボードのファームウェアのアップデートを行います	番号指定
SVOUSB20_FpgaUpdate	SVO ボードの FPGA コンフィグレーション・データのアップデートを行います	番号指定
SVOUSB20_GpioAcc	SVO ボード上の GPIO ポートを制御します	番号指定
SVOUSB20_MemClear	SVO ボード上の画像格納領域をゼロクリアします	番号指定
SVOUSB20_PllUpdate	SVO ボードの PLL データのアップデートを行います	番号指定

(※)番号指定とは、SVO ボードのボード番号を引数に必須であることを示します。SVO ボードのボード番号は SVOUSB20\_EnumDevice API で取得することができます。

### 3.3. SVO 画像出力ライブラリ API リファレンス

#### 3.3.1. SVOUSB20\_Init

**API** SVOUSB20\_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**  
 DWORD SVOUSB20\_Init( void );

**戻り値**  
 SVOUSB20\_RET\_NORMAL 正常終了

**備考**  
 ・必ず最初に呼び出して下さい。

#### 3.3.2. SVOUSB20\_EnumDevice

**API** SVOUSB20\_EnumDevice

**機能** SVO ボードの接続台数及び接続された SVO ボードの番号を接続台数分取得します

**プロトタイプ**  
 DWORD SVOUSB20\_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**  
 SVOUSB20\_RET\_NORMAL 正常終了  
 SVOUSB20\_RET\_ERROR\_DEVOPEN SVO ボードが接続されていないか、SVO ボードの電源が入っていません  
 SVOUSB20\_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 SVOUSB20\_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 ボードのボード番号を取得できます。

#### 3.3.3. SVOUSB20\_End

**API** SVOUSB20\_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 SVOUSB20\_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. SVOUSB20\_Open

API SVOUSB20\_Open

機能 指定の SVO 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 SVOUSB20_Open (
    ULONG        ulBoardNum        // SVO ボード番号を指定します
);
```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_ERROR_DEVOPEN	SVO ボードが接続されていないか、SVO ボードの電源が入っていません
SVOUSB20_RET_ERROR_MULTIOOPEN	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
SVOUSB20_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 ボード番号は、SVOUSB20\_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. SVOUSB20\_Close

API SVOUSB20\_Close

機能 指定の SVO 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 SVOUSB20_Close (
    ULONG        ulBoardNum        // SVO ボード番号を指定します
);
```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_RET_ERROR_PARAMETER	引数に間違いがあります

## 備考

・SVO ボード番号は、SVOUSB20\_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. SVOUSB20\_ReadReg

API SVOUSB20\_ReadReg

機能 指定の SVO ボードの SVO ハードウェア・レジスタから値を読み込みます

## プロトタイプ

```

DWORD SVOUSB20_ReadReg (
    ULONG    ulBoardNum    // SVO ボード番号を指定します
    ULONG    ulAddr,      // SVO ハードウェア・レジスタ・アドレス
    ULONG    ulDataWicth, // データバイト幅(1、2、4)
    PULONG   pulReadData  // 読み込んだ値を格納するポインタ
);

```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_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 ボード番号は、SVOUSB20\_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 ハードウェア・レジスタ・アドレスは「SVO-01 ハードウェア・レジスタ表」をご覧ください。

## 3.3.7. SVOUSB20\_WriteReg

API SVOUSB20\_WriteReg

機能 指定の SVO ボードの SVO ハードウェア・レジスタへ値を書き込みます

## プロトタイプ

```

DWORD SVOUSB20_WriteReg (
    ULONG    ulBoardNum    // SVO ボード番号を指定します
    ULONG    ulAddr,      // SVO ハードウェア・レジスタ・アドレス
    ULONG    ulDataWicth, // データバイト幅(1、2、4)
    ULONG    ulWriteData  // 書き込み値
);

```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_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 ボード番号は、SVOUSB20\_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 ハードウェア・レジスタ・アドレスは「SVO-01 ハードウェア・レジスタ表」をご覧ください。



**3.3.8. SVOUSB20\_BusReset**

API SVOUSB20\_BusReset

機能 USB バス上にバス・リセットを発行します

**プロトタイプ**

```
DWORD SVOUSB20_BusReset (
    ULONG    ulBoardNum    // SVO ボード番号を指定します
);
```

**戻り値**

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_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**

・本 API の後には必ず SVOUSB20\_DeviveReset API を発行して下さい。

**3.3.9. SVOUSB20\_DeviceReset**

API SVOUSB20\_DeviceReset

機能 SVO ボードを初期化し、アイドル状態にします

**プロトタイプ**

```
DWORD SVOUSB20_DeviceReset (
    ULONG    ulBoardNum    // SVO ボード番号を指定します
);
```

**戻り値**

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_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 ボード番号は、SVOUSB20\_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。

**3.3.10. SVOUSB20\_MemClear**

API SVOUSB20\_MemClear

機能 SVO ボード上の画像格納領域をゼロクリアし、画像データなし状態とします

**プロトタイプ**

```
DWORD SVOUSB20_MemClear (
    ULONG    ulBoardNum    // SVO ボード番号を指定します
);
```

**戻り値**

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_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 ボード番号は、SVOUSB20\_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。

## 3.3.11. SVOUSB20\_GetBoardStatus

API SVOUSB20\_GetBoardStatus

機能 SVO 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 SVOUSB20_GetBoardStatus (
    ULONG        ulBoardNum,    // SVO 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

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_NORMAL_OUT	正常終了(出力中)
SVOUSB20_RET_NORMAL_STOP	正常終了(停止中)
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_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;     // 現在の動作モード
                                // bit31 : エラー状態
                                // bit30-05 : 予約(0)
                                // bit04 : SDRAM に画像データあり
                                // bit03-00 :
                                // 0x8 アップデート中
                                // 0x2 画像データ受信
                                // 0x1 出力中
                                // 0x0 アイドル中

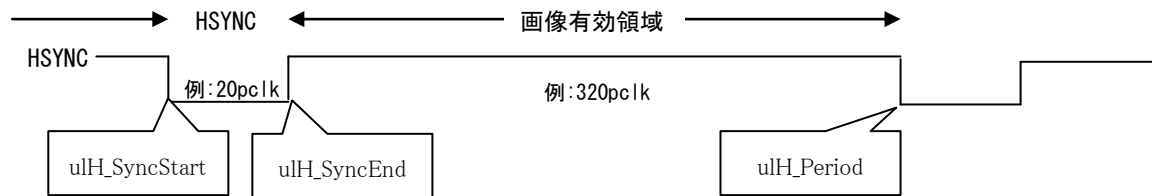
    ULONG    ulHWEdition;    // FPGA のエディション番号
    ULONG    ulH_Period;     // H 有効 PCLK 数(備考)
    ULONG    ulH_SyncStart;  // H 同期開始 PCLK 数(備考)
    ULONG    ulH_SyncEnd;    // H 同期終了 PCLK 数(備考)
    ULONG    ulDIP_SW2;     // DIP スイッチ S2 モニター
    ULONG    ulV_Period;     // V 有効ライン数(備考)
    ULONG    ulV_SyncStart;  // V 同期開始ライン数(備考)
    ULONG    ulV_SyncEnd;    // V 同期終了ライン数(備考)
    ULONG    ulSyncInfo;     // 同期信号設定情報
                                // bit31-02 : 予約(0)
                                // bit01 : ピクセルクロック極性(0: ↑、1: ↓)
                                // bit00 : 同期信号極性(0:Low Active、1:High Active)

    ULONG    ulSyncMonitor;  // 同期信号モニター
                                // bit31-08 : 予約(0)
                                // bit07 : HSYNC (Output)
                                // bit06 : VSYNC (Output)
                                // bit05-03 : 予約(0)
                                // bit02 : ターゲットボード電源情報(0:OFF、1:ON)
                                // bit01 : 予約(0)
                                // bit00 : ターゲットボードリセット情報(0:解除、1:リセット中)

    ULONG    ulSdramInfo;    // SDRAM 容量(128 or 256 単位 MB)
    ULONG    ulBoardVer;     // SVI ボードバージョン番号(SVO-02:02h、SVO-03:03h)
    ULONG    ulPllClock;     // PLL Clock 設定値(単位 KHz)
} GET_BSTATUS, *PGET_BSTATUS;
```

## 備考

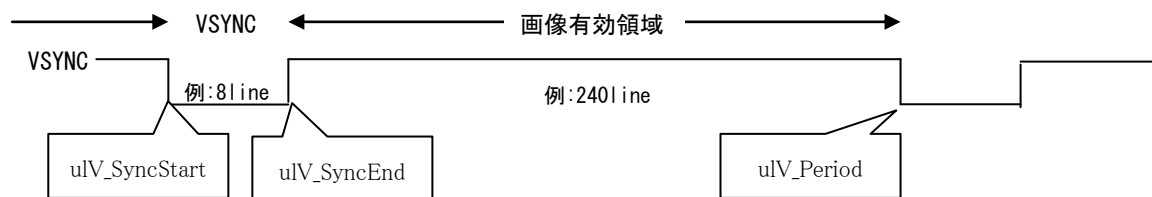
- SVO ボード番号は、SVOUSB20\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ulH\_Period (H 有効 PCLK 数)、ulH\_SyncStart (H 同期開始 PCLK 数)、ulH\_SyncEnd (H 同期終了 PCLK 数) は下図を参照してください。



上記の場合の設定値:

```
ulH_Period = 340;           // H 有効 PCLK 数 320 + 20
ulH_SyncStart = 0;         // H 同期開始 PCLK 数
ulH_SyncEnd = 20;          // H 同期終了 PCLK 数
```

- ulV\_Period (V 有効ライン数)、ulV\_SyncStart (V 同期開始ライン数)、ulV\_SyncEnd (V 同期終了ライン数) は下図を参照してください。



上記の場合の設定値:

```
ulV_Period = 248;           // V 有効ライン数 240 + 8
ulV_SyncStart = 0;          // V 同期開始ライン数
ulV_SyncEnd = 8;            // V 同期終了ライン数
```

## 3.3.12. SVOUSB20\_GetOutputStatus

API SVOUSB20\_GetOutputStatus

機能 画像出力時の情報(フレーム番号、トータル番号など)を取得します

## プロトタイプ

```

DWORD SVOUSB20_GetOutputStatus (
    ULONG          ulBoardNum,    // SVO ボード番号を指定します
    PULONG         pulFrameNum,   // 現在出力中のフレーム番号を格納するポインタ
                                // アイドル時:0
                                // オフライン出力時:0~9999
    PULONG         pulTotalNum    // オフライン出力開始からの出力フレーム総数を格納するポインタ
);

```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_NORMAL_OUT	正常終了(出力中)
SVOUSB20_RET_NORMAL_STOP	正常終了(停止中)
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_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 ボード番号は、SVOUSB20\_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 は主にオフライン出力時に使用してください。pulFrameNum と戻り値により、出力状態を確認することが可能です。
- 強制停止後のフレーム番号は停止した時のフレーム番号を格納します。

## 3.3.13. SVOUSB20\_SetParam

API SVOUSB20\_SetParam

機能 SVO ボードへ出力設定情報を通知します

## プロトタイプ

```
DWORD SVOUSB20_SetParam (
    ULONG        ulBoardNum,    // SVO ボード番号を指定します
    PSET_PARAM    pParam        // 出力設定情報構造体のポインタ
);
```

## 戻り値

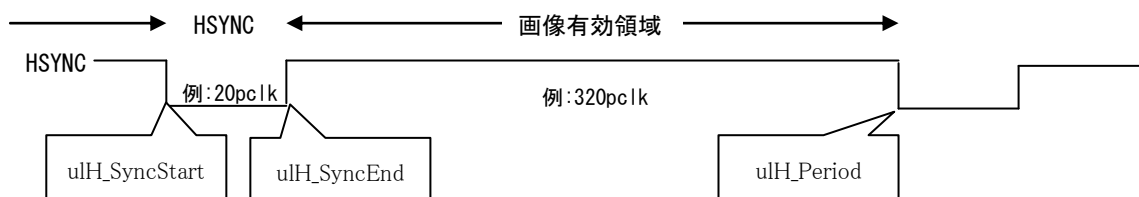
SVOUSB20\_RET\_NORMAL                      正常終了  
 SVOUSB20\_RET\_ERROR\_NOOPEN              オープンされていません  
 SVOUSB20\_RET\_ERROR\_PARAMETER          引数に間違いがあります  
 その他                                  Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

## 構造体

```
typedef struct _SET_PARAM {
    ULONG    ulH_Period;           // H 有効 PCLK 数(備考)
    ULONG    ulH_SyncStart;        // H 同期開始 PCLK 数(備考)
    ULONG    ulH_SyncEnd;          // H 同期終了 PCLK 数(備考)
    ULONG    ulExternalTrigger;    // 外部トリガー設定 (0:OFF、1:ON)
    ULONG    ulV_Period;           // V 有効ライン数(備考)
    ULONG    ulV_SyncStart;        // V 同期開始ライン数(備考)
    ULONG    ulV_SyncEnd;          // V 同期終了ライン数(備考)
    ULONG    ulSyncInfo;           // 同期信号設定情報
                                     // bit31-02 : 予約 (0)
                                     // bit01   : ピクセルクロック極性 (0: ↑、1: ↓)
                                     // bit00   : 同期信号極性 (0:Low Active、1:High Active)
} SET_PARAM, *PSET_PARAM;
```

## 備考

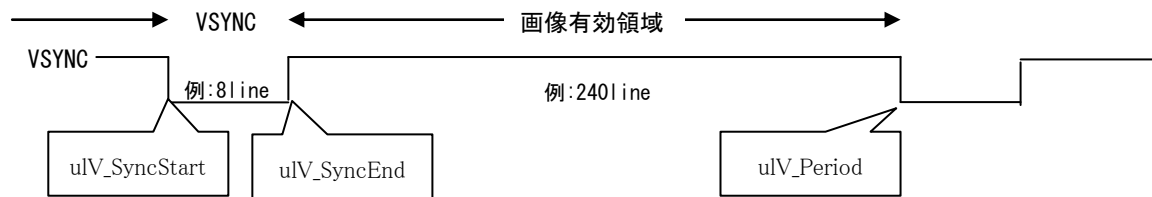
- SVO ボード番号は、SVOUSB20\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- ulH\_SyncStart (H 同期開始 PCLK 数)、ulV\_SyncStart (V 同期開始ライン数) には 0 のみを指定してください。
- ulH\_Period (H 有効 PCLK 数)、ulH\_SyncStart (H 同期開始 PCLK 数)、ulH\_SyncEnd (H 同期終了 PCLK 数) は下図を参照してください。



上記の場合の設定値:

```
ulH_Period = 340;           // H 有効 PCLK 数 320 + 20
ulH_SyncStart = 0;          // H 同期開始 PCLK 数
ulH_SyncEnd = 20;           // H 同期終了 PCLK 数
```

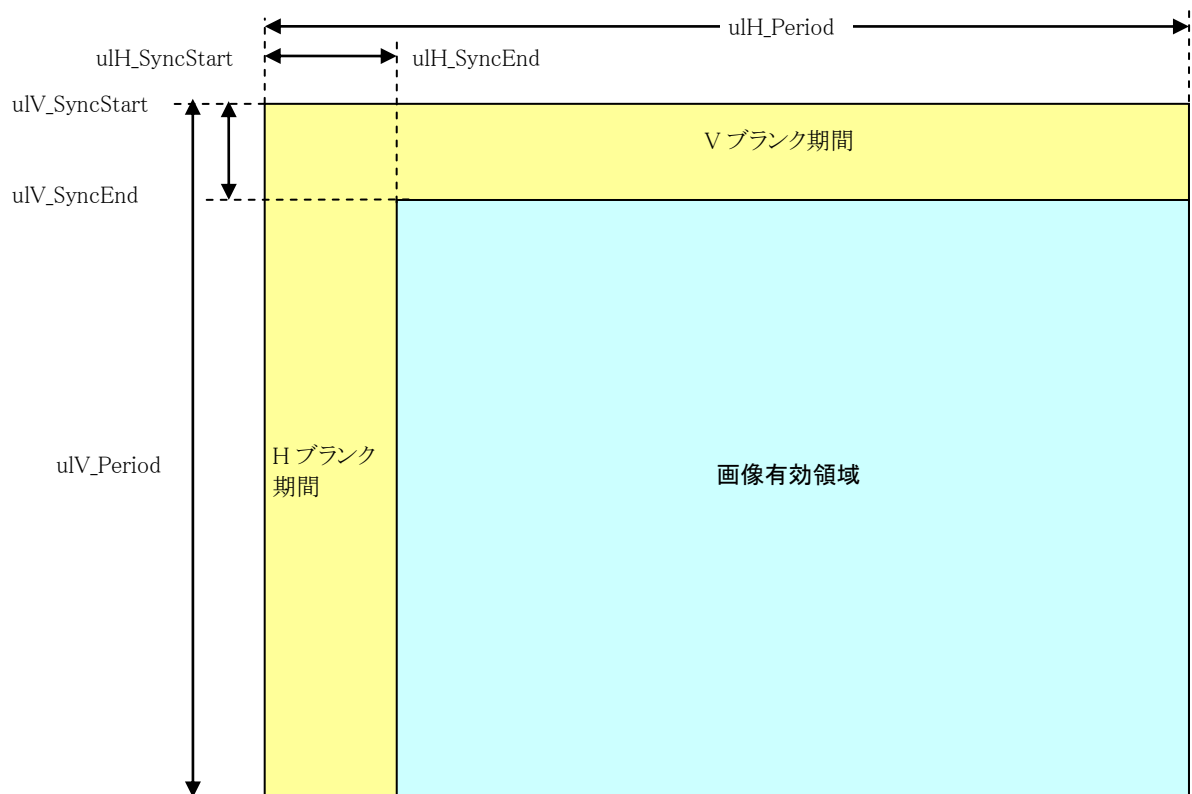
•ulV\_Period (V 有効ライン数)、ulV\_SyncStart (V 同期開始ライン数)、ulV\_SyncEnd (V 同期終了ライン数) は下図を参照してください。



上記の場合の設定値:

```
ulV_Period = 248;           // V 有効ライン数 240 + 8
ulV_SyncStart = 0;         // V 同期開始ライン数
ulV_SyncEnd = 8;          // V 同期終了ライン数
```

上記設定項目を下図にて示します。



## 3.3.14. SVOUSB20\_OfflineOutputStart

API SVOUSB20\_OfflineOutputStart

機能 SVO 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 SVOUSB20_OfflineOutputStart (
    ULONG    ulBoardNum,        // SVO 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       // 繰り返し回数(0～9999)
);

```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_NORMAL_OUT	正常終了(出力中)
SVOUSB20_RET_NORMAL_STOP	正常終了(停止中)
SVOUSB20_RET_ERROR_NODATA	出力する画像データが格納されていません
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_RET_ERROR_PARAMETER	引数に間違いがあります
SVOUSB20_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 ボード番号は、SVOUSB20\_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 0 の場合は、SVOUSB20\_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、SVOUSB20\_OfflinePictureSend API にて画像データを SVO ボードに送信、格納する必要があります。

## 3.3.15. SVOUSB20\_OfflineOutputStop

API SVOUSB20\_OfflineOutputStop

機能 SVO 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 SVOUSB20_OfflineOutputStop (
    ULONG    ulBoardNum,        // SVO ボード番号を指定します
);

```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_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 ボード番号は、SVOUSB20\_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、SVOUSB20\_OfflineOutputStart API にてオフライン出力を開始する必要があります。

## 3.3.16. SVOUSB20\_OfflinePictureSend

API SVOUSB20\_OfflinePictureSend

機能 SVO ボードへ画像データを送信します

## プロトタイプ

```

DWORD SVOUSB20_OfflinePictureSend (
    ULONG        ulBoardNum,    // SVO 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        ulReserve;     // 予約(0)
    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

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_RET_ERROR_PARAMETER	引数に間違いがあります(ポインタが NULL)
その他	Win32API エラー (bit31-28:EH, bit27-0:GetLastError 戻り値)

## 備考

- SVO ボード番号は、SVOUSB20\_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は SVOUSB20\_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。また画像データのバイト数は 64 で割り切れるように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 SVOUSB20\_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。

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。



## 3.3.17. SVOUSB20\_FirmwareUpdate

API SVOUSB20\_FirmwareUpdate

機能 SVO ボードのファームウェアのアップデートを行います

## プロトタイプ

```

DWORD SVOUSB20_FirmwareUpdate (
    ULONG    ulBoardNum,        // SVO ボード番号を指定します
    LPVOID    lpFirmData,       // ファームウェア・アップデート格納ポインタ
    ULONG    ulDataSize        // アップデータバイトサイズ(4 の倍数)
);

```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_RET_ERROR_PARAMETER	引数に間違いがあります(ポインタが NULL)
SVOUSB20_RET_ERROR_NOTIDLE	SVO ボードがアイドル状態ではありません
SVOUSB20_RET_ERROR_FMUPDATE_T	書き込み終了待ちでタイムアウトが発生しました
その他	Win32API エラー (bit31-28:EH, bit27-0:GetLastError 戻り値)

## 備考

- ・SVO ボード番号は、SVOUSB20\_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 実行中は、他の API は使用しないで下さい
- ・本 API 実行前に SVO ボードがアイドル状態であることを確認して下さい。

## 3.3.18. SVOUSB20\_FpgaUpdate

API SVOUSB20\_FpgaUpdate

機能 SVO ボードの FPGA コンフィグレーション・データのアップデートを行います

## プロトタイプ

```

DWORD SVOUSB20_FpgaUpdate (
    ULONG    ulBoardNum,        // SVO ボード番号を指定します
    LPVOID    lpFpgaData       // FPGA コンフィグレーション・アップデート格納ポインタ
    ULONG    ulDataSize;       // アップデータバイトサイズ(4 の倍数)
);

```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_RET_ERROR_PARAMETER	引数に間違いがあります(ポインタが NULL)
SVOUSB20_RET_ERROR_NOTIDLE	SVO ボードがアイドル状態ではありません
SVOUSB20_RET_ERROR_FPUPDATE_T	書き込み終了待ちでタイムアウトが発生しました
その他	Win32API エラー (bit31-28:EH, bit27-0:GetLastError 戻り値)

## 備考

- ・SVO ボード番号は、SVOUSB20\_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 実行中は、他の API は使用しないで下さい
- ・本 API 実行前に SVI ハードウェアがアイドル状態であることを確認して下さい。

**3.3.19. SVOUSB20\_GpioAcc**

API SVOUSB20\_GpioAcc  
 機能 SVO ボード上の GPIO ポートを制御します  
 プロトタイプ  
 DWORD SVOUSB20\_GpioAcc (  
     ULONG ulBoardNum, // SVO 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  
 SVOUSB20\_RET\_NORMAL 正常終了  
 SVOUSB20\_RET\_ERROR\_PARAMETER 引数に間違いがあります  
 SVOUSB20\_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 ボード番号は、SVOUSB20\_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 ポートのみ制御なります。

**3.3.20. SVOUSB20\_GetVersion**

API SVOUSB20\_GetVersion  
 機能 SVOUSB20.DLL のバージョン情報を取得します  
 プロトタイプ  
 DWORD SVOUSB20\_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  
 SVOUSB20\_RET\_NORMAL 正常終了  
 SVOUSB20\_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

```

## 3.3.21. SVOUSB20\_PllUpdate

API SVOUSB20\_PllUpdate

機能 SVO ボードの PLL データのアップデートを行います

## プロトタイプ

```

DWORD SVOUSB20_PllUpdate (
    ULONG    ulBoardNum,        // SVO ボード番号を指定します
    LPVOID    lpFirmData,       // PLL データ格納ポインタ
    ULONG    ulDataSize        // アップデータバイトサイズ(4 の倍数)
);

```

## 戻り値

SVOUSB20_RET_NORMAL	正常終了
SVOUSB20_RET_ERROR_NOOPEN	オープンされていません
SVOUSB20_RET_ERROR_PARAMETER	引数に間違いがあります(ポインタが NULL)
SVOUSB20_RET_ERROR_NOTIDLE	SVO ボードがアイドル状態ではありません
SVOUSB20_RET_ERROR_FMUPDATE_T	書き込み終了待ちでタイムアウトが発生しました
その他	Win32API エラー (bit31-28:EH、bit27-0:GetLastError 戻り値)

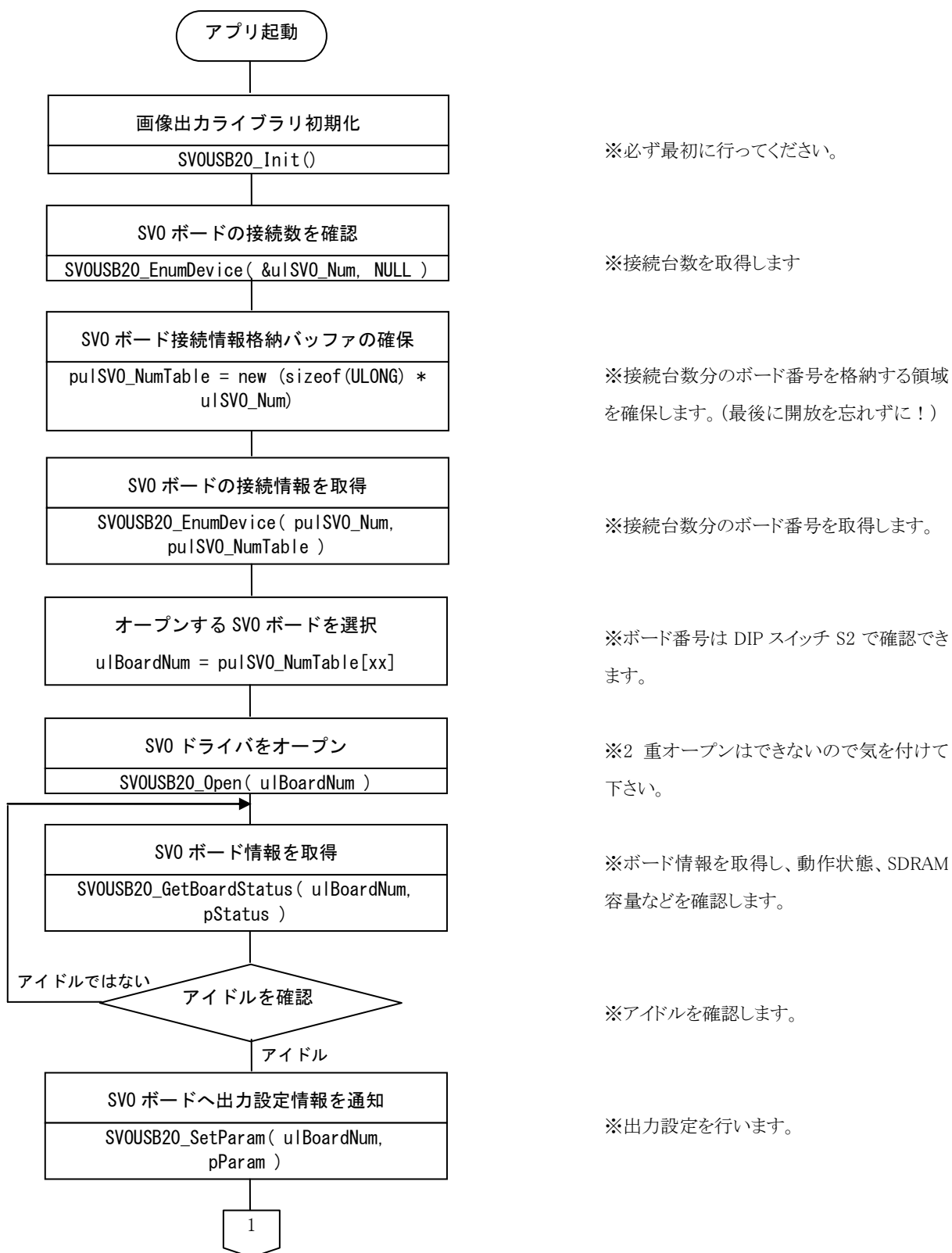
## 備考

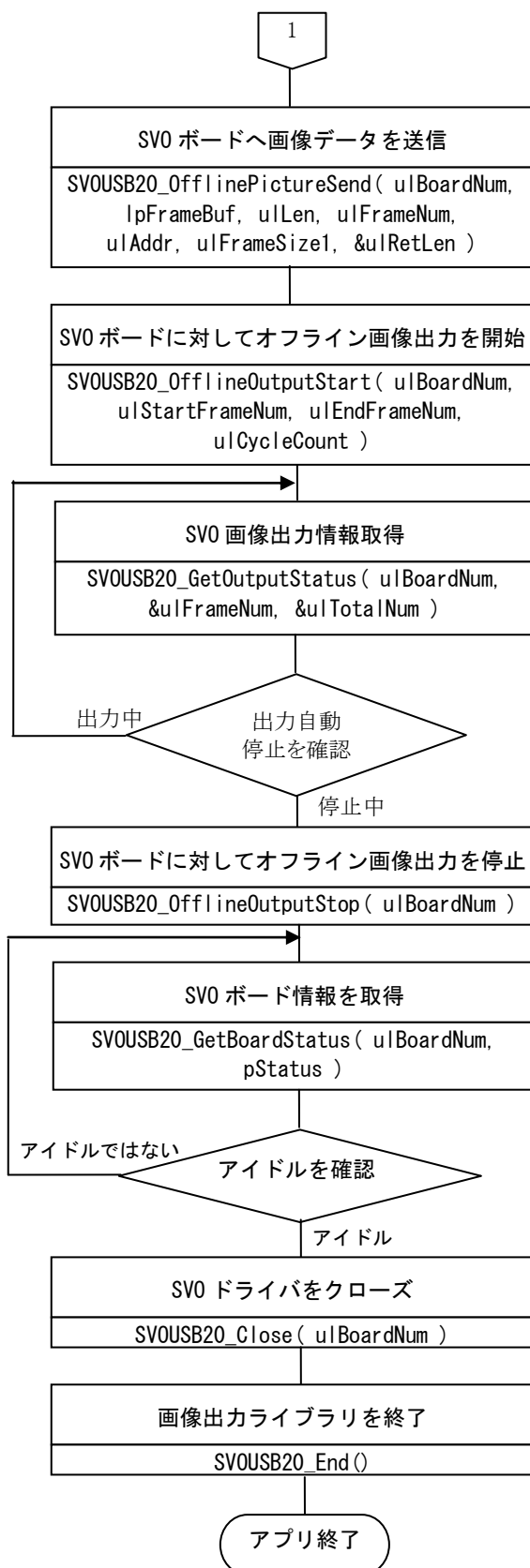
- SVO ボード番号は、SVOUSB20\_EnumDevice API にて取得したテーブル内のいずれかの番号を指定してください。
- 本 API 実行中は、他の API は使用しないで下さい
- 本 API 実行前に SVO ボードがアイドル状態であることを確認して下さい。
- 本APIは、SVO-02用ファームウェア V1.03から有効なAPIです。またPLLデータを変更するためにSVO-02用ブートROMがPLL変更に対応している必要があります。

## 3.3.22. エラーコード一覧

エラーコード名	コード	内容
SVOUSB20_RET_NORMAL	0x00000001	正常終了
SVOUSB20_RET_NORMAL_OUT	0x00000002	正常終了(出力中)
SVOUSB20_RET_NORMAL_STOP	0x00000003	正常終了(停止中)
SVOUSB20_RET_ERROR_DEVICE	0xe0000000	Win32API エラー発生時に GetLastError の値に OR され
SVOUSB20_RET_ERROR_DEVOPEN	0xf0000001	SVO ボードが接続されていないか、電源が入っていない
SVOUSB20_RET_ERROR_MULTIOOPEN	0xf0000002	SVO ボードを 2 重にオープンしようとした
SVOUSB20_RET_ERROR_NOOPEN	0xf0000003	SVO ボードがオープンされていない
SVOUSB20_RET_ERROR_PARAMETER	0xf0000004	パラメーターエラー発生
SVOUSB20_RET_ERROR_NOTIDLE	0xf0000005	SVO ボードがアイドル状態ではない
SVOUSB20_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
SVOUSB20_RET_ERROR_FPUPDATE_T	0xf0000007	FPGA データ更新タイムアウトエラー発生
SVOUSB20_RET_ERROR_BUSY	0xf1000011	ビジーでコマンドが実行できない
SVOUSB20_RET_ERROR_NODATA	0xf1000012	SVO ボードに画像データが格納されていない
SVOUSB20_RET_ERROR_CMD_INVALID	0xf1000021	コマンドが不正である
SVOUSB20_RET_ERROR_PRM_INVALID	0xf1000022	パラメータが不正である
SVOUSB20_RET_ERROR_CMD_NOIMPLE	0xf1000023	SVO ボードでサポートしていないコマンドである
SVOUSB20_RET_ERROR_SEQ_INVALID	0xf1000023	シーケンスが不正である
SVOUSB20_RET_ERROR_UPDATE_INVALID	0xf10000e1	モジュールデータが不正である
SVOUSB20_RET_ERROR_FROMERS	0xf10000e2	フラッシュメモリーの消去に失敗した
SVOUSB20_RET_ERROR_FROMWT	0xf1000043	フラッシュメモリーの書き込みに失敗した
SVOUSB20_RET_ERROR_INTERNAL	0xf10000f1	内部エラーが発生した
SVOUSB20_RET_ERROR_RESOURCE	0xf10000f2	内部リソースが不足して処理が実行できない

## 3.3.23. オフライン出力時の画像出力ライブラリ使用例





※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 SV0USB20\_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。

