

Smart Vision Image
Software Development Kit
アプリケーション内部仕様書

V2.20

承認	承認	照査	作成

株式会社ネットビジョン

改訂履歴

版数	日付	内容	担当	備考
2.00	2009/02/03	・ SVIsam V2.0.0.0にあわせ、修正	柏木	
2.01	2011/08/24	・ SVI-03SK用SVIsam対応	町野	
2.02	2011/10/20	・ 3.5.8. Settingを変更	柏木	
2.03	2013/02/20	・ 3.5.8. Settingを変更 ・ 12.5. 12.6にI2C通信時のフローチャートを追加	柏木	
2.10	2020/05/08	・ 14. 動作環境を更新	柏木	
2.20	2021/06/11	・ 動作環境など更新	柏木	

1.	適用	5
2.	概要	5
3.	サンプルアプリケーション仕様	6
3.1.	SVISAMメイン画面機能	8
3.2.	ステータス表示エリア	9
3.2.1.	<i>FrameRate</i>	9
3.2.2.	<i>Size</i>	10
3.2.3.	<i>FrameNo</i>	10
3.2.4.	<i>Zoom</i>	10
3.3.	ユーザーコマンドエリア	11
3.4.	コントロールエリア	12
3.5.	コマンドエリア	13
3.5.1.	<i>SingleMonitor</i>	14
3.5.2.	<i>MultiMonitor</i>	14
3.5.3.	<i>Recording</i>	14
3.5.4.	<i>CameraPowerOff</i>	15
3.5.5.	<i>ColorBar</i> . . .	16
3.5.6.	<i>BitmapSave</i> . . .	17
3.5.7.	<i>WFM&VecSco</i>	18
3.5.8.	<i>Setting</i> . . .	20
3.6.	インフォメーションエリア	22
4.	モニタリング	25
4.1.	シングルモニタリング	25
4.2.	マルチモニタリング	27
4.3.	モニタリングデータ	28
5.	レコーディング	30
5.1.	レコーディングデータ	31
6.	ダブルバッファ	33
7.	YUV->RGB変換	34

8.	設定ファイル	35
9.	波形表示について	36
9.1.	波形モニタ.....	36
9.2.	ベクタースコープ.....	39
10.	エラーメッセージ	40
10.1.	カメラデバイスに関するエラー.....	40
10.2.	設定条件に関するエラー.....	42
10.3.	ファイルアクセスに関するエラー.....	44
10.4.	アプリケーションの動作に関するエラー.....	45
11.	各種動作フロー	46
11.1.	シングルモニタリング動作フロー.....	47
11.2.	マルチモニタリング動作フロー.....	48
11.3.	モニタリングスレッドフロー.....	49
11.4.	レコーディング動作フロー.....	50
11.5.	レコーディングスレッドフロー.....	51
11.6.	インフォメーション動作フロー.....	52
11.7.	波形表示動作フロー.....	53
11.8.	シングルモニター正常動作シーケンス.....	54
11.9.	シングルモニターONTIMER正常動作シーケンス(モニタリング・グラフ表示なしの処理).....	55
11.10.	シングルモニターSETFRAME正常動作シーケンス(モニタリング・グラフ表示なしの処理).....	56
11.11.	モニタリングスレッド正常動作シーケンス.....	57
12.	SVI API各種フロー	58
12.1.	モニタリングモード時の画像入力ライブラリ使用例.....	58
12.2.	レコーディングモード時の画像入力ライブラリ使用例.....	60
12.3.	SPIによるコマンド送信時の画像入力ライブラリ使用例.....	62
12.4.	SPIによるコマンド受信時の画像入力ライブラリ使用例.....	63
12.5.	I2Cによるコマンド送信時の画像入力ライブラリ使用例.....	64
12.6.	I2Cによるコマンド受信時の画像入力ライブラリ使用例.....	65
12.7.	複数台のSVI-09を接続する場合のAPI使用例.....	66
12.8.	複数台のSVI-09を使用する場合の使用環境について.....	66
13.	ソースファイル一覧	67
14.	動作環境	68

15. 開発環境 68

1. 適用

本設計仕様書は Smart Vision Image 用 Software Development Kit に適用します。

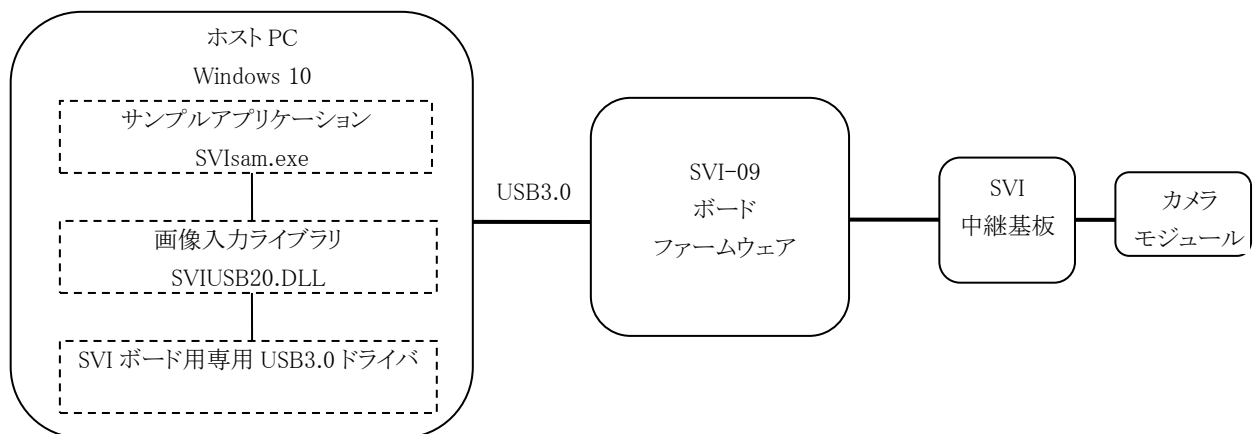
2. 概要

Smart Vision Image - Software Development Kit (以降：SVI-SDK) とはカメラ・モジュールの評価を目的とした Windows 上のソフトウェアと SVI ボード及びファームウェアと SVI 中継基板から構成されます。カメラモジュールは顧客が評価しようとするモジュールになります。

本 SDK はあらかじめ Windows 及び VS2008/VC++ について基礎的な知識を有していることを前提に内容を記述します。VC++ 等については別に参考書等をご覧ください。

本書では、サンプルアプリケーション (SVIsam.exe) の内部仕様について記述します。

【図 2.1】 SVI-SDK システム構成図



このシステムは、SVI ボードによって制御されたカメラ・モジュールの画像データを Windows10 搭載ホスト PC に USB3.0 インターフェースで取り込むことが可能なシステムです。サンプルアプリケーション SVIsam.exe は 32bit 版/64bit 版があります。

【表 2.1】 SVI-SDK 構成一覧表

ソフトウェア	サンプルアプリケーション	SVIsam.exe
	画像入力ライブラリ	SVIUSB20.DLL
	SVI ボード用専用 USB ドライバ	Vendor 版
	SVI ボード制御用ファームウェア	sxfwxxx.bin
ハードウェア	SVI ボード	SVI-09
	SVI ボード制御用 FPGA データ	sxfpgaxxx.bin
	SVI 中継基板(※)	MIPI ボード、LVDS ボードなど

3. サンプルアプリケーション仕様

サンプルアプリケーション（以降 SVIsam）は、SVI 画像入力ライブラリの使用方法を説明するためのソース公開を前提とした Windows アプリケーションです。

SVI 画像入力ライブラリの各 API を使用し、シングル画面モニタリング、マルチ画面モニタリング、レコーディング、SPI コントロールをひとつのアプリケーションで実現します。SPI のコントロールは本書に記述されるパラメータファイルを送信することで容易に行うことができます。言語は英語（アメリカ）を対象とします。

実現するアプリケーションの機能を表 3.1 に示します。

【表 3.1】 SVIsam 機能一覧

機能名称	機能動作
シングルモニタリング	画像表示エリアいっぱいモニタリングで取得した画像を描画します。画像が表示エリアを超える場合、表示エリアの右及び下にあるスクロールバーを用いて画像をスクロールさせます。 シングルモニタリングの詳細は「3.5.1. SingleMonitor」、 「4.1 シングルモニタリング」をご覧ください。
マルチモニタリング	画像表示エリアを 9 分割し、描画機能を有効にした箇所にたいしてモニタリングで取得した画像を指定した位置より切り出して描画します。画像表示エリアのスクロールバーは無効となります。 マルチモニタリングの詳細は「3.5.2. MultiMonitor」、 「4.2 マルチモニタリング」をご覧ください。
レコーディング	SVI ボードで画像の蓄積（レコーディング）を行い、蓄積終了後アプリケーションでデータの取り込みを行いフレームを描画します。 レコーディングの詳細は「3.5.3. Recording」、 「5. レコーディング」をご覧ください。
Color bar	描画画像上をマウスカーソルで指定された場合にその座標の RGB 及び YUV 値を表示します。値の更新はマウスカーソルが移動した際に行われます。 ColorBar の詳細については「3.5.5. ColorBar」をご覧ください。
Information	レコーディング以外の動作中にカメラのインフォメーションをリアルタイム及びマニュアルで表示します。 Information の詳細については「3.6. Information」をご覧ください。

	下さい。
BitmapSave	<p>モニタリングまたはレコーディングで描画されている画像をビットマップで保存します。</p> <p>BitmapSave については「3.5.6. BitmapSave」をご覧ください。</p>
CameraPowerOn/Off	<p>SVI ボードのレジスタを書き換えることによりカメラパワーの On/Off を切り替えます。</p> <p>CameraPowerOn/Off については「3.5.4. CameraPowerOff」をご覧ください。</p>
モニタリング切り出し位置設定	<p>モニタリングでの画像の切り出し位置を設定します。マルチモニタリングでの設定はシングルモニタリングで設定したカメラ切り出し位置及びサイズに対してマルチモニタリングで設定した位置から固定幅でサイズを切り出します。</p> <p>切り出し設定については「3.5.7. Setting」をご覧ください。</p>
カメラコントロール	<p>SPI によるカメラモジュールへの設定ファイルの送信、指定スレーブアドレス、サブアドレスのリード/ライトを行います。</p> <p>カメラコントロールエリアについては「3.4 コントロールエリア」をご覧ください。</p>
パラメータファイル送信	<p>カメラモジュールへあらかじめ指定したパラメータファイルを送信します。</p> <p>パラメータファイル送信については「3.3 ユーザーコマンドエリア」をご覧ください。</p>
ウェブフォーム表示 ベクタースコープ表示	<p>画像表示エリアいっぱいモニタリングで取得した画像と、その画像の X 輝度、Y 輝度波形グラフとベクタースコープ描画します。画像が表示エリアを超える場合、表示エリアの右及び下にあるスクロールバーを用いて画像をスクロールさせます。波形表示についてはシングルモニター時のみ有効となります。そのためシングルモニター表示中以外は表示されません。</p> <p>Wave Form & Vector Scope の詳細は「3.5.7. WFM&VecSco」をご覧ください。</p>

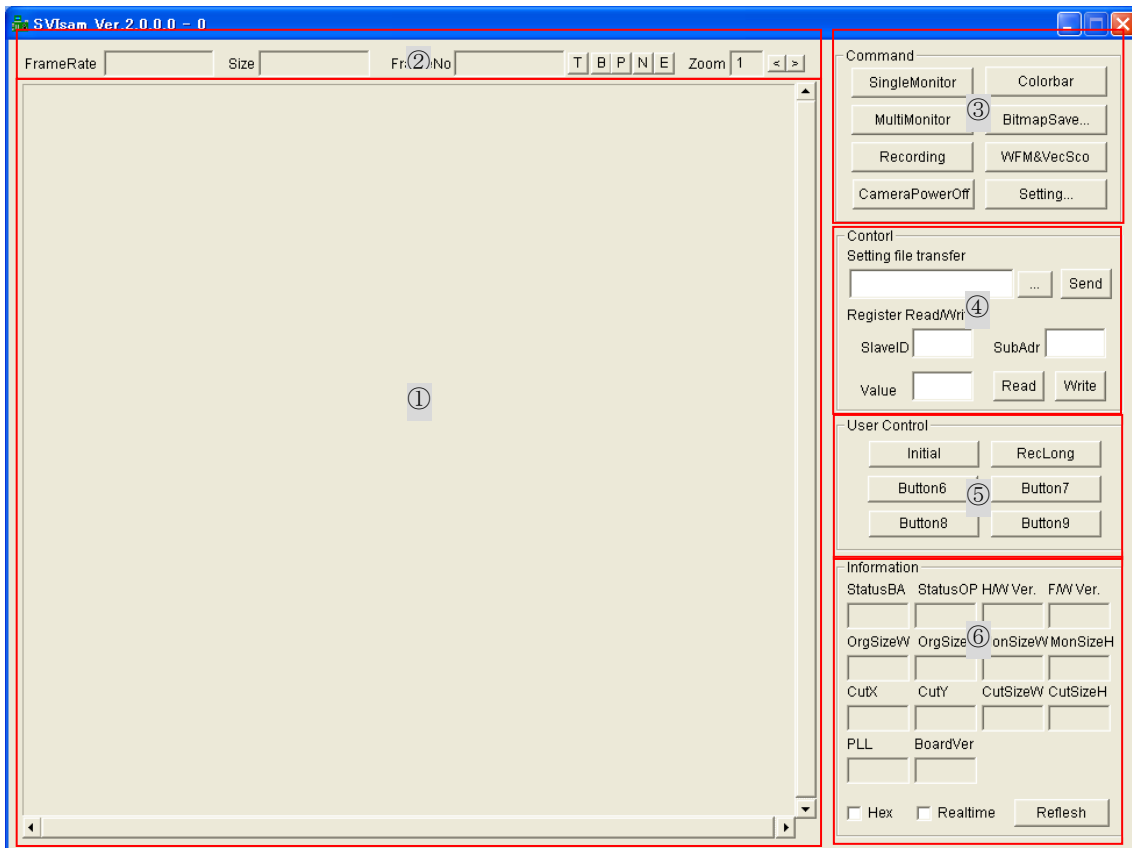
3.1. SVIsam メイン画面機能

図 3.1 に SVIsam の操作画面を示します。

SVIsam の操作画面は、5 つの領域から構成されます。

※タイトルバーには” SVIsam” の他に本アプリケーションのバージョン番号とボード番号も表示します。

【図 3.1】 SVIsam 操作画面





Area		
名称	番号	内容
画像表示エリア	①	モニタリング、レコーディングで取得したフレームを表示します。
ステータス表示エリア	②	モニタリング、レコーディングの情報を表示と、表示倍率の変更、レコーディングフレームの操作を行います。
コマンドエリア	③	モニタリング、レコーディングなど各機能を実行します。
コントロールエリア	④	カメラへのパラメータファイルの送信、レジスタへの Read/Write を行います。

ユーザーコマンドエリア	⑤	カメラモジュールへユーザー独自のパラメータファイルを送信することが簡単に行えます。また、モニタリング応用昨日として長時間録画機能を行えます。
インフォメーションエリア	⑥	SVI ボードのステータスを表示します。

3.2. ステータス表示エリア

このエリアにはモニタリング時、レコーディング時のステータス表示、表示フレームの倍率変更、レコーディングフレーム操作を行います。

【図 3.2】 ステータス表示エリア

Control	
名称	内容
FrameRate	モニタリング時のフレームレートを少数第 2 位まで表示 ※読み取り専用
Size	画像表示エリアの描画している画像のサイズ ※読み取り専用
FrameNo  フレーム操作ボタン	レコーディング時のフレーム番号/総フレーム数 レコーディングデータ内のフレームの操作を行います ※読み取り専用
Zoom  Zoom 率変更ボタン	拡大、縮小倍率の表示と倍率変更 拡大は 8 倍、4 倍、2 倍、縮小は 1/8 倍、1/4 倍、1/2 倍が行えます。 初期値は 1 倍表示です。 ※読み取り専用

3.2.1. FrameRate

「FrameRate」はモニタリングにおける 1 秒間あたりの表示フレームレート/カメラ出力フレームレートを表示します。シングルモニタリングまたはマルチモニタリングを開始するとフレームレートの表示も自動的に開始されます。シングルモニタリング、マルチモニタリングが停止されるとフレームレートも自動的に空欄となります。

単位は Frame / second [f/s] です。

レコーディング画像表示時には無効（空欄）となります。

表示フレームレートは 4 秒間で表示した総フレーム数を 4 で割って求めています。

カメラ出力フレームレートは SVI ボードがビデオ同期信号を検知したカメラからのフレームレートです。

3.2.2. Size

「Size」は現在画像表示エリアに表示されているモニタリング画像またはレコーディング画像のサイズを1フレーム単位で幅 x 高さを表示します。画像が表示されている間 Size も表示されません。カメラからの出力画像サイズではなく、SVI ボードでの切り出し後のサイズとなります。切り出し設定は Setting ボタンで行います。

3.2.3. FrameNo

「FrameNo」はレコーディング時の表示フレーム番号／総フレーム数を表示します。モニタリング画像表示時は無効（空欄）となります。レコーディングデータが内部で保持される間はフレーム数を表示します。

レコーディングで取り込んだフレーム数が2フレーム以上の場合は、右にあるフレーム操作ボタンが有効になります。T、B、P、N、E ボタンはそれぞれ TopFrame、BackFrame、Play、NextFrame、EndFrame を意味します。各ボタンの動作は以下のとおりです。

【表 3.2】 フレーム操作

ボタン名	操作名	動作
T	TopFrame	1 番目のフレームを表示します。
B	BackFrame	前のフレームを表示します。
P	Play	フレームを連続再生します。フレーム再生中はボタンのラベルが「S」（Stop）に変わり、動作が停止処理へ変更します。最後のフレーム表示後自動的に停止します。
N	NextFrame	次のフレームを表示します。
E	EndFrame	最後のフレームを表示します。

これらの動作は内部でフレームカウンタを操作することにより実現します。

3.2.4. Zoom

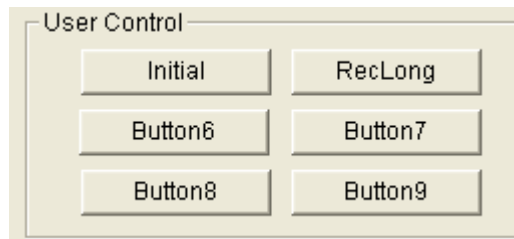
「Zoom」は現在の表示倍率を表示します。左右のボタンにより表示倍率を変更します。範囲は8倍、4倍、2倍、1倍、1/2倍、1/4倍、1/8倍となります。画像の表示はシングルモニタリング、マルチモニタリング、レコーディングのフレームそれぞれに対して有効です。アプリケーション起動時は1倍とします。

拡大処理は等倍の RGB 画像から描画に必要なエリアを切り出し、切り出したエリアのデータを拡大することにより画像の描画を行います。

3.3. ユーザーコマンドエリア

カメラモジュールへパラメータファイルを送信する機能を提供します。また RecLong というモニタリング応用機能も提供します。Button6 から Button9 の名称は INI ファイルで変更できます。このエリアの画面イメージは以下のとおりです。

【図 3.3】 ユーザーコマンドエリア



Control	
名称	内容
Initial SPI/I2C 送信	初期化用パラメータファイル“initial.dat”を送信します。このファイルはサンプルを参考にユーザーでご用意下さい。
RecLong モニタリング	モニタリング応用機能でモニタリングフレームをハードディスクへ長時間保存する機能を提供します。長時間保存時のファイルフォーマットは弊社規定の FRM ファイルというものです。
Button6 - 9 SPI/I2C 送信他	各ボタンに対応するパラメータファイルを送信することが出来ます。パラメータファイル名はボタン名称を同一でファイル名の最後に“.dat”を付加したものとなります。上図の場合、Button6 は“Button6.dat”となります。パラメータファイルの書式は Initil.dat と同様です。こちらのファイルもユーザーでご用意下さい。 Button6 から 9 のユーザーコマンドはパラメータファイルの送信だけでなく、ユーザーで自由に機能を変更することが出来ます。サンプルとして RecLong ボタン処理をご覧下さい。

SPI/I2C によるレジスタへのアクセスについては、「11. SVI API 各種フロー」部分をご覧下さい。送信するパラメータファイルは対象となるカメラ毎に異なるため、その都度ユーザーで作成してください。パラメータファイルの作成方法については「8. 設定ファイル」をご覧下さい。

3.4. コントロールエリア

このエリアは SPI によるカメラモジュールへの設定ファイルの送信、指定スレーブアドレス、サブアドレスのリード/ライトを行います。

コントロールエリアの各処理及びパラメータファイル送信エリアの各処理は排他的に処理を行い、カメラモジュールへの同時アクセスを回避します。

【図 3.4】 Control エリア

Control	
名称	内容
Setting file transfr	送信するパラメータファイルのパスを表示または入力します。 最大入力数 : 255 最小入力数 : 0
...	ファイル指定コマンドダイアログを表示します。ダイアログで指定されたファイルパスを「Setting file transfer」へ表示します。
Send	指定されたパラメータファイル及び他設定値でカメラモジュールへの Read/Write を行います。
Register Read/Write	
SlaveID	レジスタ SlaveID。Read/Write 時未入力不可。 最大値 : 0xFF 最小値 : 0x00
Value	レジスタへの書込値及び読込値。Write 時未入力不可。 最大値 : 0xFF 最小値 : 0x00
SubAdr	レジスタサブアドレス。Read/Write 時未入力不可。 最大値 : 0xFFFF 最小値 : 0x00
Read	カメラモジュールの指定されたレジスタアドレスの読み込みを行う。読み込んだ値は「Value」へ表示する。

Write	カメラモジュールの指定されたレジスタアドレスへ書き込みを行う。書き込む値は「Value」の入力値とする。
-------	--

「...」ボタンを押すことによりファイルを開くダイアログが表示し、パラメータファイルを選択します。パラメータファイルの拡張子は「.dat」です。「SlaveID」、「Value」、「SubAdr」は16進で入力します。

各エリアへの入力はアプリケーション終了時に保存し、次回アプリケーション起動時に読み込んで設定するようにします。

「Send」ボタンを押すことにより、「Setting file transfer」に設定されているパスのパラメータファイルをカメラモジュールへ SPI または I2C 経由で送信します。パラメータファイル送信処理中は Read/Write 処理及びパラメータファイル送信エリアの処理を行えません。

「Read」ボタンを押すことにより、指定スレーブアドレス、サブアドレスのレジスタをリードして「Value」エディットボックスへ表示します。Read 処理中はパラメータファイル送信/Write 処理及びパラメータファイル送信エリアの処理を行えません。

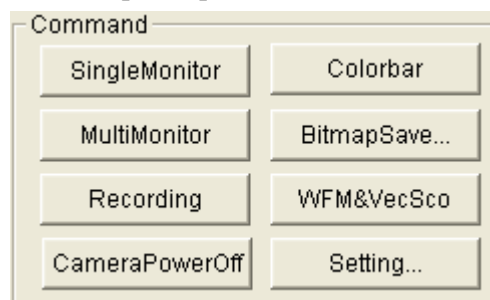
「Write」ボタンを押すことにより、指定スレーブアドレス、サブアドレスのレジスタへ「Value」エディットボックスの値をライトします。Write 処理中はパラメータファイル送信/Read 処理及びパラメータファイル送信エリアの処理を行えません。

パラメータファイル送信、Read 及び Write 処理はワーカスレッドを新たに作成し、そのスレッド内で処理を実行します。ウィンドウへの必要な処理についてはメインスレッド内で行います。SPI または I2C によるレジスタへのアクセスについては「11. SVI API 各種フロー」をご覧ください。エラーが発生した場合はエラーメッセージを表示し、処理を終了します。

3.5. コマンドエリア

このエリアは SVIsam のコマンドを配置しており下図のように 8 つのコマンドを表示します。

【図 3.5】 Command エリア



Control	
名称	内容
SingleMonitor	シングルモニタリング開始/停止トグルボタン

MultiMonitor	マルチモニタリング開始/停止トグルボタン
Rcording	レコーディング開始/停止トグルボタン
CameraPowerOff/On	カメラ電源 On/Off トグルボタン
ColorBar…	カラーバーダイアログ表示トグルボタン
BitmapSave…	ビットマップファイル保存ダイアログ表示ボタン
WFM&VecSco	X 輝度、Y 輝度波形とベクトルスコープの開始/停止トグルボタン
Setting…	設定ダイアログ表示ボタン

3.5.1. SingleMonitor

ボタンを押すことにより1画面のモニタリングが開始され、画像表示エリアにモニタリング画像を動画表示します。

「SingleMonitor」コマンドが開始されるとボタンが凹みます。モニタリングの停止は「SingleMonitor」ボタンを押すか、「MultiMonitor」ボタン、「Recording」ボタン、「CameraPowerOff」ボタンを押すことにより自動的に停止し、そのコマンドが実行されます。シングルモニタリングが停止するとボタンが凸状態へ戻ります。

シングルモニタリングの動作については「4.1 シングルモニタリング」をご覧ください。

3.5.2. MultiMonitor

ボタンを押すことにより多画面によるモニタリングが開始され、画像表示エリアにモニタリング画像を最大9画面動画表示します。「MultiMonitor」コマンドが開始されるとボタンが凹みます。モニタリングの停止は「MultiMonitor」ボタンを押すか、「SingleMonitor」ボタン、「Recording」ボタン、「CameraPowerOff」ボタンを押すことにより自動的に停止し、そのコマンドが実行されます。マルチモニタリングが停止するとボタンが凸状態へ戻ります。

マルチモニタリングの動作については「4.2 マルチモニタリング」をご覧ください。

3.5.3. Recording

ボタンを押すことによりレコーディングを開始し、レコーディング画像を画像表示エリアに表示します。「Recording」コマンドが開始されるとボタンが凹みます。レコーディングはレコーディング画像が表示されると自動的に停止しボタンが凸状態へ戻ります。

レコーディングの動作については「5. レコーディング」をご覧ください。

3.5.4. CameraPowerOff

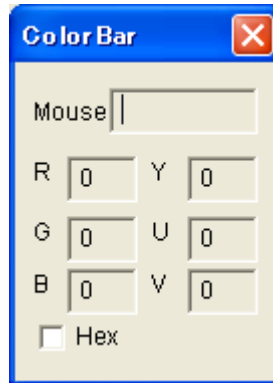
ボタンを押すことにより、SVI ボードのレジスタを書き換えることによりカメラへの電源断が実行されます。コマンド実行後、ボタン文字列が「CameraPowerOn」になります。「CameraPowerOn」ボタンを押すことにより、カメラへの電源投入が実行されます。コマンド実行後、ボタン文字列が「CameraPowerOff」になります。

アプリケーション起動時にもレジスタの読み込みを行いカメラへの電源が供給されているかを確認します。

3.5.5. ColorBar...

ボタンを押すことにより、「ColorBar」ダイアログボックスがモードレス表示されます。図 3.6 の「ColorBar」ダイアログボックスが表示されている間は、「ColorBar...」ボタンが凹みます。「ColorBar...」ボタンを「x」ボタンにより閉じると、「ColorBar...」ボタンが凸状態へ戻ります。

【図 3.6】 ColorBar



Control	
名称	内容
Position	マウスカーソルポジションを表示 表示形式 [X x Y] ※読み取り専用
Y	マウスカーソルポジションの Y 値 ※読み取り専用
U	マウスカーソルポジションの U 値 ※読み取り専用
V	マウスカーソルポジションの V 値 ※読み取り専用
R	マウスカーソルポジションの R 値 ※読み取り専用
G	マウスカーソルポジションの G 値 ※読み取り専用
B	マウスカーソルポジションの B 値 ※読み取り専用
Hex	値表示切り替え。値表示を 10/16 進で切り替えを行います。
x	ダイアログを閉じます

画像表示エリア内のマウスカーソルポジションの座標、色成分値がマウスカーソルの移動と連動して表示されます。カーソルポジションは画像表示エリア内で表示されているフレーム上でのみ有効です。それ以外の場所をカーソルが指定している場合はフレーム上で最後に取得したカラー情報の表示を維持します。

「Hex」チェックボックスのチェックが On の場合はカラー値を 16 進で、Off の場合は 10 進で表示します。「Hex」の状態は INI ファイルへ保存され、次回起動時にも有効になります。

YUV カラー情報は RGB カラーデータから YUV データを計算式で求め表示します。RGB から YUV の計算式には以下の式を使用します。

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = -0.169R - 0.331G + 0.500B$$

$$V = 0.500R - 0.419G - 0.081B$$

[カーソル位置のカラー値取得方法]

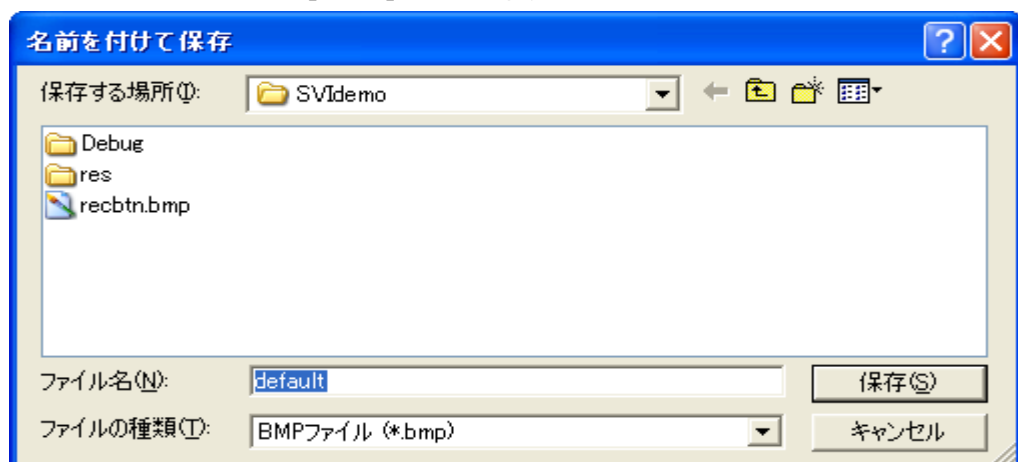
画面表示エリアに描画されている画像は内部ではビットマップとして保持されているため、バッファ内では上下が反転して保存されています。このため、カーソル位置とスクロールバーの位置から別途計算し正しい位置を求めます。

3.5.6. BitmapSave...

ボタンを押すことにより、名前をつけてファイルを保存ダイアログボックスが表示され、現在画像表示エリアに表示されている画像をビットマップ保存します。保存に使用されるデータは RGB バッファに保持されているデータを使用します。保存に使用されたファイル名は次回保存時にも有効とします。この機能はアプリケーション終了後も有効とします。

保存できるファイルの種類はビットマップファイルのみとし、表示されるファイルもビットマップのみとします。

【図 3.7】 ファイル指定コマンドダイアログ



※ RGB バッファについては「ダブルバッファ」を参照してください。

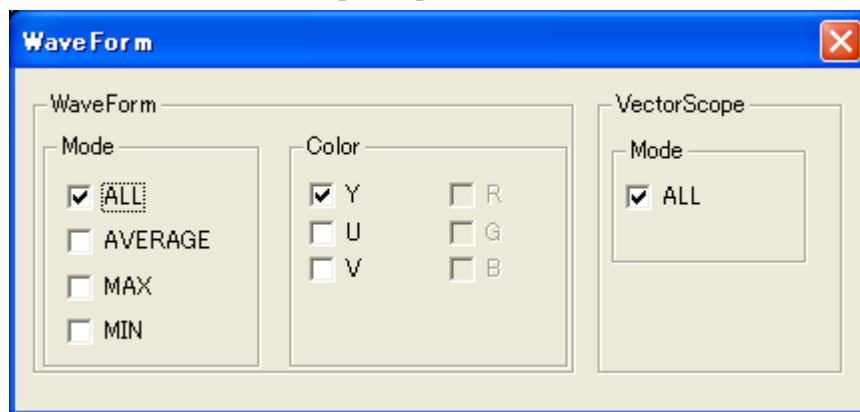
3.5.7. WFM&VecSco

SingleMonitor 開始中にボタンを押すことにより、ダイアログが表示され、同時に現在表示中のカメラ画像に対する、X・Y 輝度波形とベクタースコープを表示します。そのため、SingleMonitor 開始中以外には表示することはできません。

「WFM&VecSco」コマンドが開始されるとボタンが凹みます。グラフ表示の停止は「WFM&VecSco」ボタンを押すか、「MultiMonitor」ボタン、「Recording」ボタン、「CameraPowerOff」ボタンを押すことにより自動的に停止し、そのコマンドが実行されます。波形表示が停止するとボタンが凸状態へ戻りません。

ダイアログの選択により表示内容を変化させることができます。ダイアログの選択による動作の詳細は「9. 波形表示について」をご覧ください。

【図 3.8】 WFM&VecSco



Control	
名称	内容
WaveForm	波形モニタの表示内容を選択します
Mode	波形モニタの表示データの扱い方を選択します
	ALL カメラ画像の全てのデータを波形モニタ上に表示する
	AVERAGE カメラ画像の 1 LINE 毎の平均値を波形モニタに表示する
	MAX カメラ画像の 1 LINE 毎の最大値を波形モニタに表示する
	MIN カメラ画像の 1 LINE 毎の最小値を波形モニタに表示する

Color	波形モニタに表示する表示色を選択します Y 輝度情報を波形モニタに表示します U U情報を波形モニタに表示します V V情報を波形モニタに表示します
VectorScope Mode	ベクトルスコープの表示内容を選択します ベクトルスコープの表示データの扱い方を選択します ALL カメラ画像の全てのデータをベクトルスコープ上に表示する

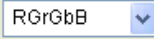
3.5.8. Setting...

ボタンを押すことにより「CommandSetting」ダイアログボックスが表示されます。

【図 3.9】 CommandSetting

Control	
名称	内容
Cutout basic setting	モニタリングの切り出し位置及びサイズを設定します モニタリングの切り出し位置 (X) CutX ※奇数、負数不可 モニタリングの切り出し位置 (Y) CutY ※負数不可 モニタリングの切り出しサイズ (横方向) CutSizeW ※奇数、0、負数不可 モニタリングの切り出しサイズ (縦方向) CutSizeH ※0、負数不可
MultiMonitor cutout setting	マルチモニタリング動作時「Cutout basic setting」 で設定した取り込みフレームから画像表示エリア毎に

<p>チェックボックス</p> <p>Cut1X ~ Cut9X</p> <p>Cut1Y ~ Cut9Y</p> <p>Cut1SizeW ~ Cut9SizeW</p> <p>Cut1SizeH ~ Cut9SizeH</p>	<p>切り出す位置の設定を行います</p> <p>画像表示エリアの有効/無効切り替え</p> <p>取り込みフレームからの切り出し位置 (X) ※奇数、負数不可</p> <p>取り込みフレームからの切り出し位置 (Y) ※負数不可</p> <p>取り込みフレームからの切り出しサイズ (横方向) アプリケーション側で計算するので表示のみ ※入力不可</p> <p>取り込みフレームからの切り出しサイズ (縦方向) アプリケーション側で計算するので表示のみ ※入力不可</p>
<p>Recording setting</p> <p>Recording Frame Count</p>	<p>レコーディングの録画フレーム数</p> <p>レコーディングの録画フレーム数 ※入力有効範囲[1~500]</p>
<p>Monitoring setting</p> <p>Double Buffer</p> <p>Ring Buffer</p> <p>SAV/EAV</p>	<p>モニタリングのバッファモードを設定します</p> <p>SVI ボード上の SDRAM を 2 つの領域に分割し取り込みを行います。画サイズが不定の時使用します。</p> <p>SVI ボード上の SDRAM を最大 32 の領域に分割し取り込みを行います。画サイズが一定またはモニタリング開始から先頭数フレームを取りこぼしたくない時に使用します。</p> <p>チェック ON で SAV/EAV コードで画像を取り込みます。 チェック OFF で VSYNC/HSYNC で画像を取り込みます。</p>
<p>Picture Type</p> <p>UYVY</p> <p>YUY2</p> <p>RGB565</p> <p>UYVY-16</p> <p>YUY2-16</p> <p>RGB565-16</p> <p>RAW-8bit</p>	<p>取り込む画像データのタイプを指定します</p> <p>UYVY (YUV422) の並びと判断します</p> <p>YUY2 (YUV422) の並びと判断します</p> <p>RGB565 形式と判断します (GB→RG の順)</p> <p>16bit 入力の UYVY (YUV422) の並びと判断します</p> <p>16bit 入力の YUY2 (YUV422) の並びと判断します</p> <p>16bit 入力の RGB565 形式と判断します (GB→RG の順)</p> <p>RAW-8bit の並びと判断します</p>

RAW-10bit RAW-120bit RAW-DPE 	RAW-10bit の並びと判断します RAW-120bit の並びと判断します RAW-8bit, RAW-10bit, RAW-12bit データを DPE して表示するかのチェック。チェック ON で DPE 後 RGB 表示、チェック OFF でグレー表示となります。グレー表示は上位 bit が 8bit を表示します ベイヤーデータの並びを指定します RGrGbB, GrRbGb, GbBRGr, BGrGrR から選択
CommunicationSetting I2C SPI 16bit SubAddress	センサーの制御方式を選択します I2C 通信で行います SPI 通信で行います I2C 通信時、サブアドレスが 16bit の時チェックします
Set	設定の更新を行いダイアログを閉じる
x	設定の変更を破棄しダイアログを閉じる

画面は 3 つのグループで構成されており、画面上部がシングルモニタリング及びマルチモニタリング動作時のモニタリング切り出し設定、画面中部がマルチモニタリング動作時の画面毎の On/Off 及びモニタリング切り出し位置設定、画面下部にレコーディング時のフレーム数の設定を行います。

全ての設定値は 10 進で入力します。

「Set」ボタンを押すことにより、設定を反映し、このダイアログボックスを閉じます。

「x」ボタンを押すことにより、設定を破棄し、このダイアログボックスを閉じます。

各設定可能コントロールの状態及び入力値はダイアログが閉じられるときにファイルへ保存し、次回起動時も設定を保持できることとします。

3.6. インフォメーションエリア

SVI ボードのステータス、ファームウェアバージョン、FPGA バージョン、モニタリング画像サイズ（モニタリング動作中のみ）を表示します。

【図 3.10】 Information

Information

StatusBA StatusOP HW Ver. FW Ver.

OrgSizeW OrgSizeH MonSizeW MonSizeH

CutX CutY CutSizeW CutSizeH

PLL BoardVer

Hex Realtime Refresh

Control	
名称	内容
StatusBA	基本ステータスを表示 ※読み取り専用
StatusOP	動作ステータスを表示 ※読み取り専用
H/W Ver.	ハードウェア (FPGA) バージョンを表示 ※読み取り専用
F/W Ver.	ファームウェアバージョンを表示 ※読み取り専用
OrgSizeW	カメラのオリジナル出力サイズを表示 (横方向) ※読み取り専用
OrgSizeH	カメラのオリジナル出力サイズを表示 (縦方向) ※読み取り専用
MonSizeW	カメラのモニタリング出力サイズを表示 (横方向) ※読み取り専用
MonSizeH	カメラのモニタリング出力サイズを表示 (縦方向) ※読み取り専用
CutX	カメラの切り出し位置を表示 (X) ※読み取り専用
CutY	カメラの切り出し位置を表示 (Y) ※読み取り専用
CutSizeW	カメラの切り出しサイズを表示 (横方向) ※読み取り専用
CutSizeH	カメラの切り出しサイズを表示 (縦方向)

	※読み取り専用
Hex	表示を 10/16 進切り替え
Realtime	リアルタイム更新の切り替え
Refresh	表示の更新
PLL	格納されている PLL データの周波数を表示します。単位 KHz ※SVI-03 のみ有効
BoardVer	SVI ボードのバージョンを表示します。 SVI-03 であれが “03” と表示します。 ※SVI-03 のみ有効
x	ダイアログを閉じる

ステータスの更新は「Refresh」ボタンを押されたときとリアルタイムの 2 つの方法があります。

「Realtime」ボタンのチェックが OFF になっている場合は「Refresh」ボタンが選択可能となります。「Realtime」ボタンのチェックが ON の間は 500ms 間隔でステータスを更新します。

表示は「H/W Ver.」と「F/W Ver.」以外の項目を 10 進数/16 進数に切り替えることができます。

「HEX」ボタンのチェックが ON になっている場合は 16 進数で表示されます。「HEX」ボタンのチェックが OFF になっている場合は 10 進数で表示されます。「Hex」及び「Realtime」のチェック状態は INI ファイルに保存され、次回起動時にも有効となります。

4. モニタリング

モニタリング処理はカメラモジュールから出力画像をリアルタイムに取得し、取得したフレームデータに対して YUV→RGB 変換を行い、GDI を用いて画像表示エリアへ描画を行います。画像の描画構成には 2 種類あり、ひとつがシングルモニタリング、もうひとつがマルチモニタリングです。（RAW データの場合は上位 8bit をグレイ表示します）

画像表示エリアの機能は CStatic クラスの派生クラスに画像描画機能を組み込むことにより実現します。シングルモニタリングではこの派生クラスをエリアいっぱいに配置しモニタリングデータを表示します。マルチモニタリングでは 9 つの派生クラスを 3x3 に均等に配置しモニタリングデータを各エリアで切り出して表示します。

モニタリング中に使用するバッファは取得したフレームのサイズと同じサイズを動的に確保し使用します。バッファサイズとモニタリングフレームのサイズが異なる場合は、その都度バッファを再確保します。モニタリングデータは YUV422 データのみを対象とし、RGB その他のフォーマットへの対応は行いません。

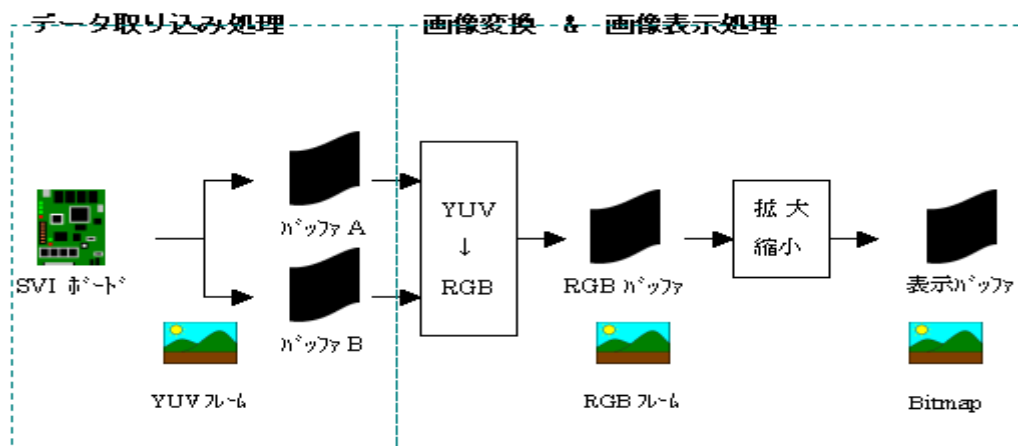
カメラデバイスへのアクセスは複数スレッドから行われるため、デバイスのオープン及びクローズは別スレッドと排他的に処理します。

4.1. シングルモニタリング

シングルモニタリングはモニタリング開始時に作成したワーカスレッドで SVI ボードから 1 フレームずつアプリ側ダブルバッファへ排他的に格納し、メインスレッドのタイマー処理で取得したフレームを YUV→RGB 変換等を行い画像表示バッファへコピーし、画像表示エリアへフレームを描画することにより、機能を実現します。

取得した画像は表示エリアの左上隅に出力画像の左上隅をあわせ描画を行います。画像が表示エリアにおさまらない場合はスクロールバーを有効にし、スクロールすることにより描画します。シングルモニタリングにおける各バッファの関係を図 4.1 に示します。

【図 4.1】 バッファ相関図 1



処理中にエラーが発生した場合はエラーメッセージを表示し、処理を終了します。表示されるエラーメッセージについては「9. エラーメッセージ」をご覧ください。

モニタリングを停止したあとモニタリング停止処理を行います。停止処理完了後に表示されているモニタリングデータは以下の処理を行うまでは有効です。

- ・ シングルモニタリング、マルチモニタリングの開始
- ・ レコーディング
- ・ アプリケーションの終了

なお画像が表示されている間はビットマップの保存、拡大縮小も行うことができます。

シングルモニタリングの際に使用される SVI の API には以下の API があります。

【表 4.1】 モニタリング関連 API

API 名称	動作
SVIUSB20_EnumDevice	接続 SVI ボードの列挙
SVIUSB20_DeviceSelect	接続する SVI ボードを指定
SVIUSB20_Open	カメラデバイスのオープン
SVIUSB20_GetStatus	カメラ詳細ステータスの取得
SVIUSB20_GetStatus2	カメラステータスの取得
SVIUSB20_MonStart	モニタリング開始
SVIUSB20_MonGetHeader	モニタリングヘッダ取得
SVIUSB20_SetRcvBuffer	バッファアドレスのセット
SVIUSB20_MonGetFrame	モニタリングフレーム取得
SVIUSB20_MonStop	モニタリング停止
SVIUSB20_Close	カメラデバイスのクローズ
SVIUSB20_DeviceRelease	接続した SVI ボードを開放

シングルモニタリングの開始から終了までの流れは「10.1 シングルモニタリング動作フロー」をご覧ください。

シングルモニタリングに関連するクラスについては「12. クラス定義一覧」をご覧ください。

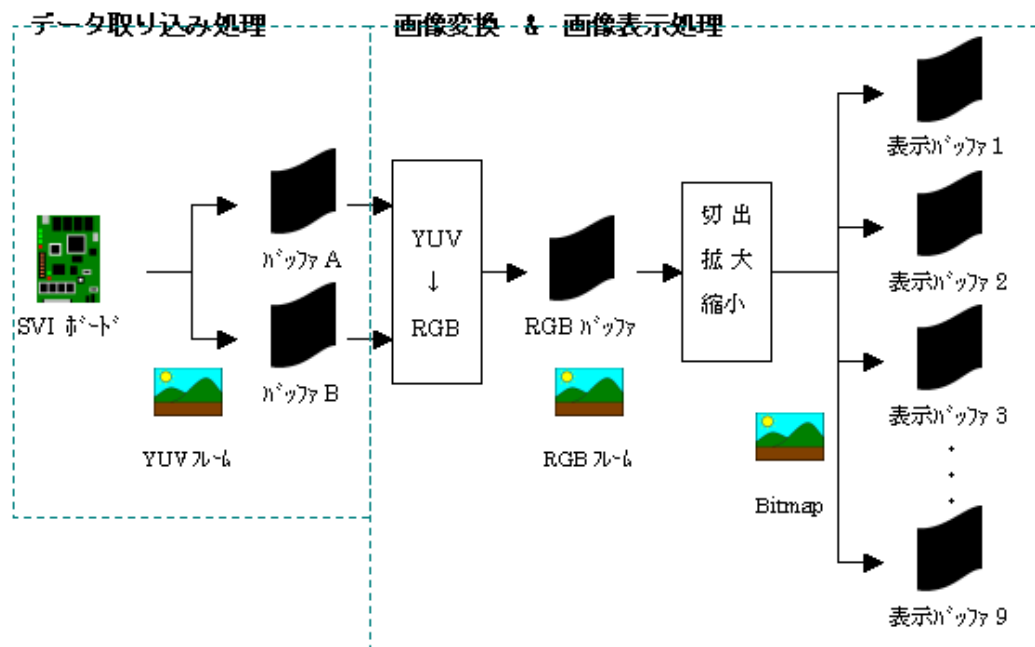
4.2. マルチモニタリング

画面構成は図 4.3 のように画像表示エリアを縦横均等に 3 等分 (9 画面) し、モニタリングで出力された画像から設定した切り出し位置及びエリアの幅で切り出した画像を描画します。また各画面の左上には画面番号と ON/OFF を表示します。

マルチモニタリングはモニタリング開始時に作成したワークスレッドで SVI ボードから 1 フレームずつアプリ側ダブルバッファへ排他的に格納し、メインスレッドのタイマー処理で取得したフレームを YUV→RGB 変換等を行い最大 9 個の画像表示バッファへコピーし、画像表示エリアへフレームを描画します。

マルチモニタリングにおける各バッファの関係を図 4.2 に示します。

【図 4.2】 バッファ相関図 2



処理中にエラーが発生した場合はエラーメッセージを表示し、処理を終了します。表示されるエラーメッセージについては「9. エラーメッセージ」をご覧ください。

モニタリングを停止したあとモニタリング停止処理を行います。停止処理完了後に表示されているモニタリングデータは以下の処理を行うまでは有効です。

- ・ シングルモニタリング、マルチモニタリングの開始
- ・ レコーディング
- ・ アプリケーションの終了

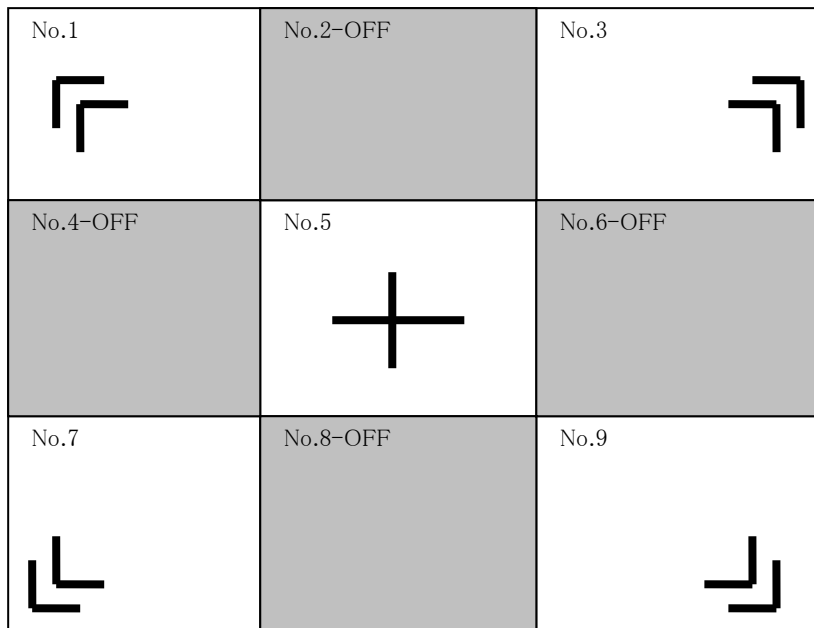
なお画像が表示されている間はビットマップの保存、拡大縮小も行うことができます。

マルチモニタリングの際に使用される SVI の API は【表 4.1】 モニタリング関連 API を御覧ください。

マルチモニタリングの開始から終了までの流れは「10.2 マルチモニタリング動作フロー」をご覧ください。

シングルモニタリングに関連するクラスについては「12. クラス定義一覧」をご覧ください。

【図 4.3】 モニタリング多画面表示



4.3. モニタリングデータ

SVI ボードから取り込まれるモニタリングデータについて説明します。

取り込まれるデータは YUV422 フォーマットとし先頭に画像左上の画素情報が格納され、順に1ラインずつ格納されています。データには YUV データのみが格納され、SYNC 情報データ等は格納されません。また、本アプリケーションにおいては YUV データの並び順は図 4.4(UYVY)のものだけを対象としております。そのためその他の並びにおいては本アプリケーションを参考にカスタマイズが必要になります。取り込んだデータのサイズ、画像の縦横サイズ等はモニタリング用のヘッダーから取得します。2画素4バイトの組み合わせとなり、不足の色差については隣り合う画素から計算によって求めます。

アプリケーションでは取得したヘッダー及びサイズからモニタリングデータを GDI で表示できるよう RGB24 ビットデータへ変換します。

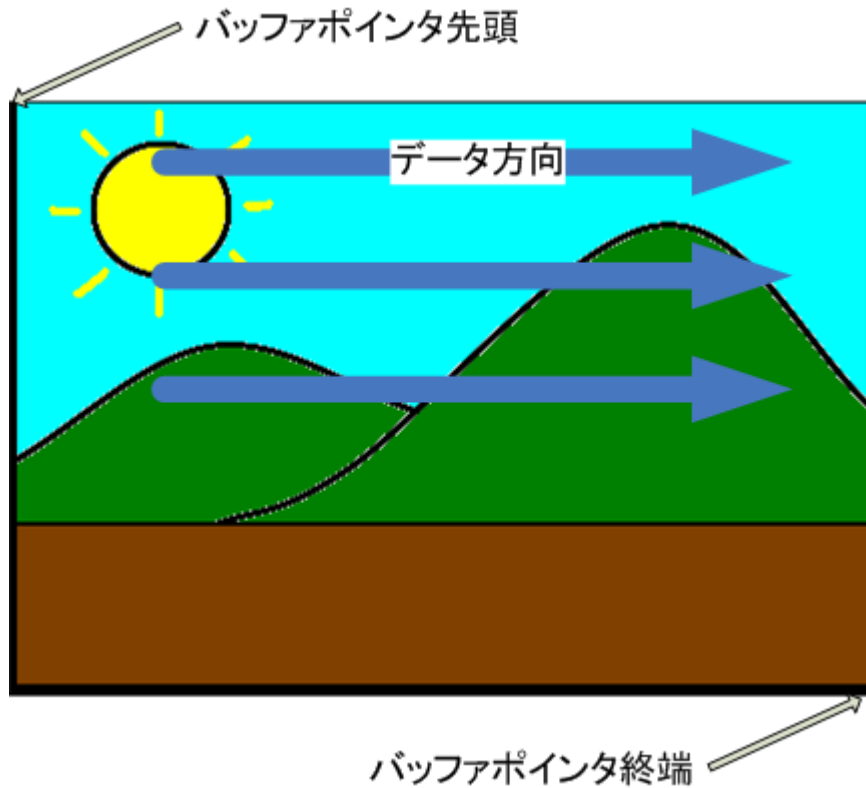
【図 4.4】 8 ビット YUV422 データ



1要素 = 1 バイト

取り込まれたデータがバッファへマッピングされた場合のイメージ図を以下に示します。

【図 4.5】 マッピングイメージ



- ※ モニタリングヘッダについては別冊「SVI 画像入力ライブラリ説明書」をご覧ください。
- ※ モニタリングの動作については「モニタリングフロー」をご覧ください。
- ※ YUV422->RGB の計算方法については「YUV->RGB 変換式」を参照してください。

以下に代表的なフレームサイズを YUV422 の場合（モニタリング）と RGB24 ビットの場合（画像表示エリア）で取り込むのに必要なバッファサイズを表でまとめましたのでご参考下さい。

【表 4.2】 バッファサイズ		
名称 (横 x 縦)	8 ビット YUV422 [BYTE]	RGB24 ビット [BYTE]
QVGA (320 x 240)	153600	230400
VGA (640 x 480)	614400	921600
XGA (1024 x 768)	1572864	2359296
SXGA (1280 x 960)	2457600	3686400
UXGA (1600 x 1200)	3840000	5760000
2M SIZE (1632 x 1224)	3995136	5992704
QXGA (2048 x 1536)	6291456	9437184
QUXGA (3200 x 2400)	15360000	23040000

5. レコーディング

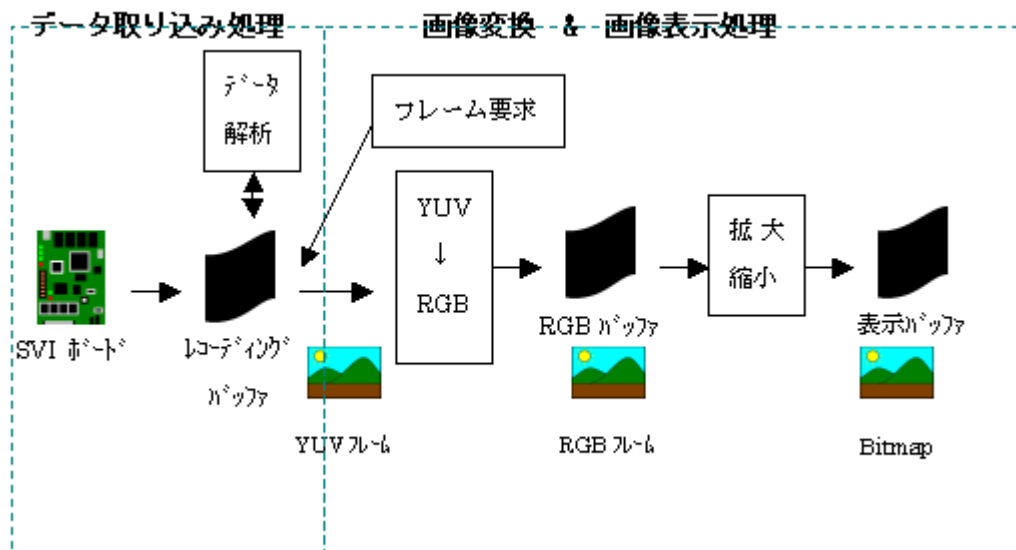
レコーディングの動作は SVI ボード（バッファサイズ 128MB）でカメラ出力を蓄積し、その蓄積したデータをアプリケーションが取得します。レコーディングにはフレーム数指定及びバイト数指定の 2 通りがあり、レコーディングに関する API の引数で指定します。本 SDK ではフレーム数によるレコーディングのみを対象とします。

データを取得したら、フレーム数と各フレームの開始位置を VSYNC/HSYNC を用いて解析します。解析終了後、1 フレーム目のデータを取り出して YUV→RGB 変換を行い、シングルモニタリングと同じ画像表示エリアへ描画します。「FrameNo」へのフレーム数の表示、「Recording」ボタンの文字を黒色に戻す処理もこのタイミングで行います。

レコーディング開始時に作成するワークスレッドで SVI API へアクセスを行い、蓄積したデータの取得を行います。

レコーディングのバッファの関係を図 5.1 に示します。

【図 5.1】 バッファ関係図 3



処理中にエラーが発生した場合はエラーメッセージを表示し、処理を終了します。表示されるエラーメッセージについては「9. エラーメッセージ」をご覧ください。

取り込んだレコーディングデータは次にあげる処理が行われるまでは有効です。

- ・ シングルモニタリング、マルチモニタリング
- ・ レコーディング
- ・ アプリケーションの終了

なお画像が表示されている間はビットマップの保存、拡大縮小も行うことができます。

表 5.1 にレコーディングに関連する SVI API を一覧で示します。

【表 5.1】 レコーディング関連 API

API 名称	動作
SVIUSB20_EnumDevice	接続 SVI ボードの列挙
SVIUSB20_DeviceSelect	接続する SVI ボードを指定
SVIUSB20_Open	カメラデバイスのオープン
SVIUSB20_GetStatus	カメラ詳細ステータスの取得
SVIUSB20_GetStatus2	カメラステータスの取得
SVIUSB20_RecStart	レコーディング開始
SVIUSB20_RecGetHeader	レコーディングヘッダ取得
SVIUSB20_SetRcvBuffer	バッファアドレスのセット
SVIUSB20_RecGetFrames	レコーディングデータ取得
SVIUSB20_RecStop	レコーディング停止
SVIUSB20_Close	カメラデバイスのクローズ
SVIUSB20_DeviceRelease	接続した SVI ボードを開放

レコーディング中に使用するバッファは取得したデータのサイズと同じサイズを動的に確保します。バッファサイズとレコーディングデータのサイズが異なる場合は、その都度バッファを再確保します。出力されるレコーディングデータのフォーマットは YUV422 データのみを対象とし、RGB を含むその他のフォーマットへの対応は行いません。レコーディングに関する処理の流れは「10.4 レコーディング動作フロー」をご覧ください。

5.1. レコーディングデータ

SVI ボードから取り込まれるレコーディングデータについて説明します。

レコーディングの開始時に指定したフレーム数蓄積されると、アプリケーションが蓄積したレコーディングデータをバッファへ取り込みます。データは図 5.2 のような 1 ピクセル 2 バイトのデータを SVI ボードで蓄積しています。各フレームに対してのサイズの情報等及びフレームを識別するためには、レコーディングデータ内に保持されている VSYNC/HSYNC 信号を見て判断します。データフォーマットは YUV422 形式、2 画素 8 バイトとなります。

【図 5.2】 ビット配列

ビット番号															
15 (MSB)	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0 (LSB)
						HSYNC	VSYNC	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]

ビット 9 に HSYNC の値、ビット 8 に VSYNC の値、ビット 7 からビット 0 に画像データが入ります。1 フレームは VSYNC が HIGH 状態の間となります。VSYNC が HIGH 状態の間で HSYNC が HIGH となっている間、画像情報が有効となっています。VSYNC が HIGH の間に HSYNC が HIGH になった数が画

像の高さとなります。

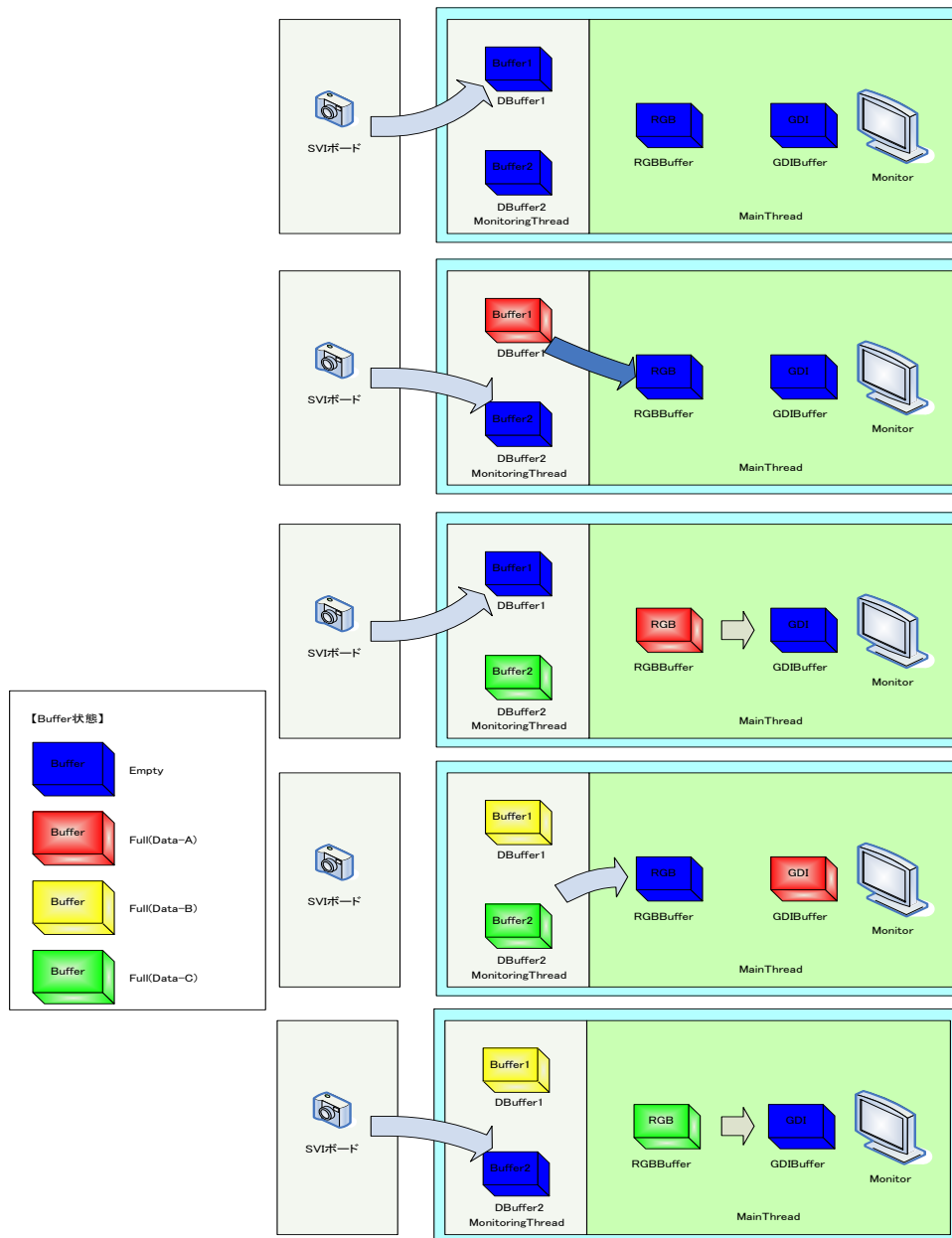
最初と最後には途中で途切れてしまっている画像が入っている可能性があります。このような場合は、その画像の次にある正常なフレームから認識するようにします。

モニタリングデータと同じように、データの先頭から画像の上部（左上）が始まります。

6. ダブルバッファ

モニタリングで SVI ボードから取り込まれるフレームはダブルバッファ動作でアプリケーションのバッファへ格納、保持されます。ダブルバッファは画像表示エリアへ表示する表示バッファへ YUV→RGB 変換・コピーされた後、空きバッファとなり、新しいフレームデータが取り込まれます。マルチモニター処理の場合は、RGB バッファまでの処理は同じとなり、GDI バッファへの切り出しコピーが最大9個のバッファに対して行われることになります。

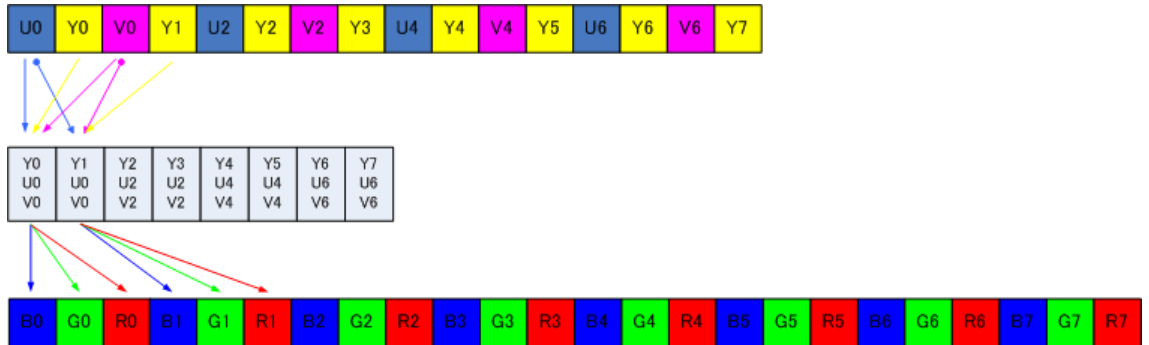
【図 6.1】ダブルバッファ動作遷移



7. YUV→RGB 変換

カメラから出力されているデータフォーマットは YUV422 形式です。この YUV422 フォーマットを RGB24 ビットへ変換するには以下の計算式を使用します。

【図 7.1】 YUV→RGB 変換図



【RGB 変換式】

$$R = Y + 1.402 * V$$

$$G = Y - 0.344 * U - 0.714 * V$$

$$B = Y + 1.772 * U$$

ビットマップファイル作成時はパディングを考慮しながら BGR の順でピクセルを格納していく。

また本アプリケーションでは YUV データの並び順は UYVY をしている。

8. 設定ファイル

SPI または I2C 通信によるカメラモジュールレジスタへの設定値書き込みをするためのテキストファイルで以下の形式で記述します。

スレーブ ID, サブアドレス, 書き込み値

Wait, ウェイト値

(Wait または wt)

スレーブ ID はデバイスのアドレスを 16 進で指定します。I2C 通信の場合、7bit のスレーブアドレスを左へ 1 ビットシフトした値で記述してください。8bit 指定です。

サブアドレスはデバイス内のサブアドレスのことであり、16 進で指定します。Setting ダイアログで I2C SubAddress チェックが ON の場合は 16bit とみなしますので注意してください。

書き込み値はレジスタへ書き込む値を 16 進で指定します。8bit 指定です。

Wait (または wt) はここでウェイト値分の時間を待つことを指定します。

ウェイト値は 10 進で、msec 単位で指定します。

1 行の終端コードはスペース、タブ、改行コードとし、それ以降のカラムは無視します。

また、セミコロン記号 “;” またはシャープ記号 “#” があると、それ以降改行までをコメントとみなし、次の行にスキップします。空行（改行のみ、スペース等を含みず）もスキップします。

設定記述例)

```
# SVI-SDK Setting file sample
# date : 2013.02.20
#
3C,00,00 ; Camera Reset
3C,01,00 # DSP Reset
#
wt,100 ; wait 100msec
#
3C,01,00 # DSP Run
3C,00,00 ; Camera Run
#
# END OF FILE
#
```

9. 波形表示について

シングルモニタリング時には、波形モニタとベクタースコープを表示することができます。このときダイアログの選択により表示内容を変化させることができます。ここではその詳細について説明します。

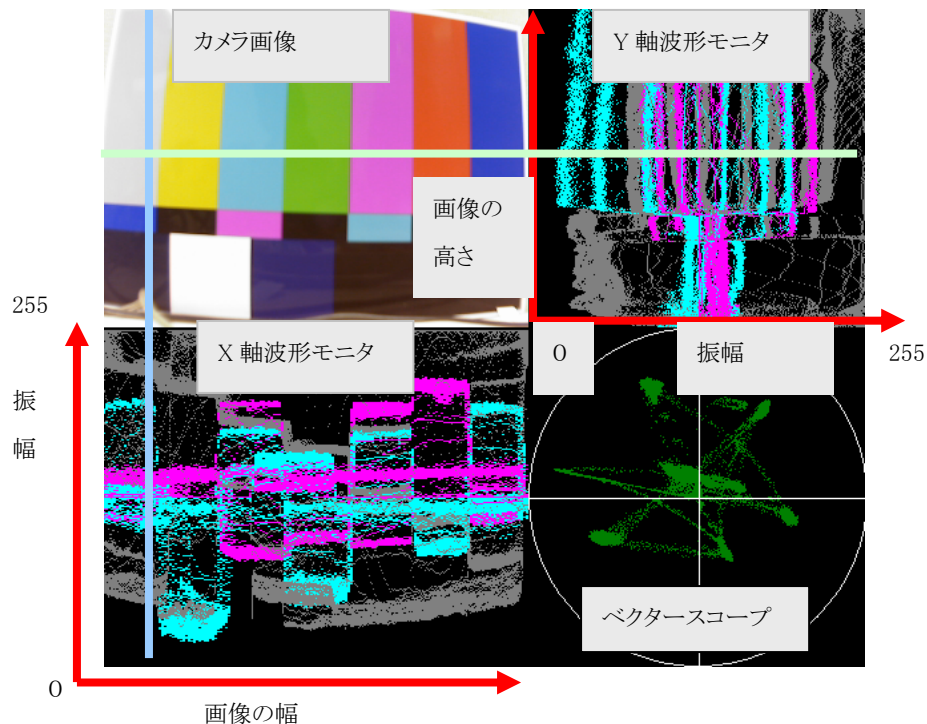
9.1. 波形モニタ

波形モニタとは一般的に「画像の明るさ」を表示するためのものになります。そして信号のノイズやひずみの観察にも利用することができます。またカラーバー信号とあわせて利用することにより明るさの調整も行うことができます。

本 SDK の波形モニタは、X 軸のカラー振幅を表示する X 軸波形モニタと Y 軸のカラー振幅を表示する Y 軸波形モニタの2つがあります。X 軸波形モニタは横軸が画像の幅、縦軸が0～255の振幅を表します。また Y 軸波形モニタは横軸が0～255の振幅を、縦軸が画像の高さを表します。そのため、X 軸波形モニタの横軸と、Y 軸波形モニタの縦軸は画像のサイズにより長さが変化することになります。

X 軸波形モニタに表示されるデータはカメラ画像の幅1画素分を取得し画素振幅に並べ替えて表示されます。Y 軸の場合には、カメラ画像の高さ1画素分が取得され画素振幅に並べ替えられます。

【図 9.1】 波形モニタ

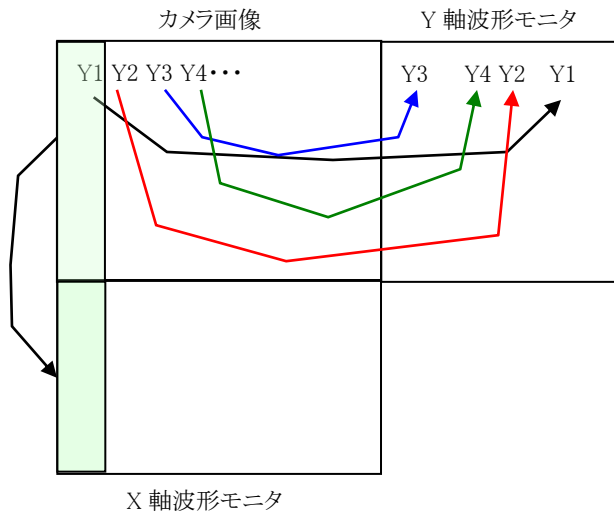


波形モニタの表示方法としては大きくわけて、「Mode」と「Color」の選択を行うことができます。また各選択は全て同時に選択することが可能になっております。

<Mode>

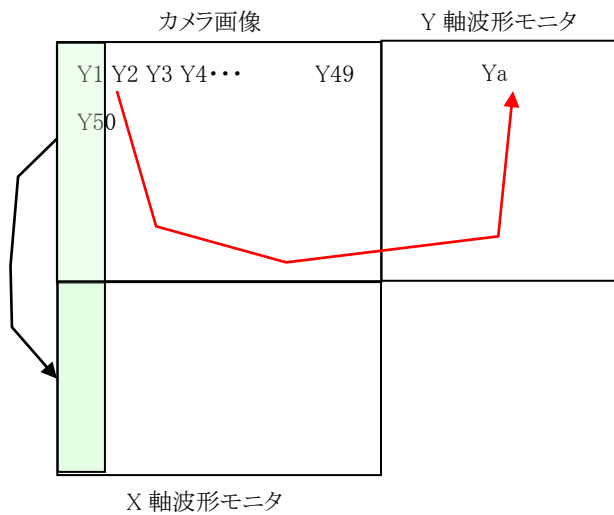
- ALL: カメラ画像の全データを波形モニタに表示します。このときカメラ画像が200 * 300のサイズであった場合には、波形モニタには60000画素の各カラー値を表示することになります。

【図 9.2】 ALL 選択時のデータの取り扱い(例:輝度値)



- AVERAGE: カメラ画像の1LINE のデータを取得し、その平均値を波形モニタに表示します。

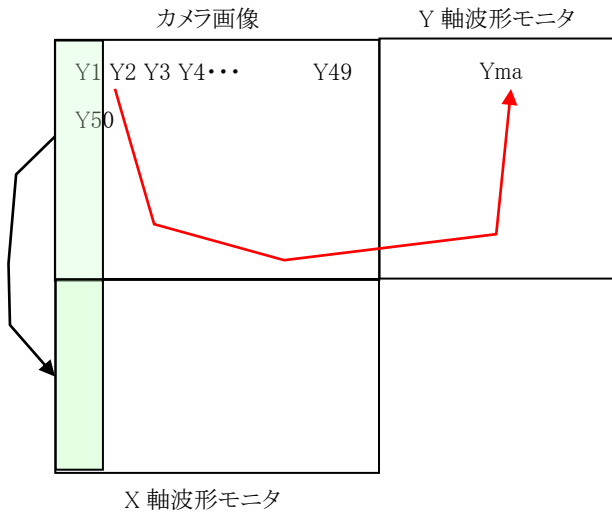
【図 9.3】 AVERAGE 選択時のデータの取り扱い(例:輝度値)



※ $Y_a = \text{Ave}(Y_1, Y_2, Y_3, \dots, Y_{49})$; 1Line 分の平均値

- MAX:カメラ画像の1LINE のデータを取得し、その最大値を波形モニタに表示します。

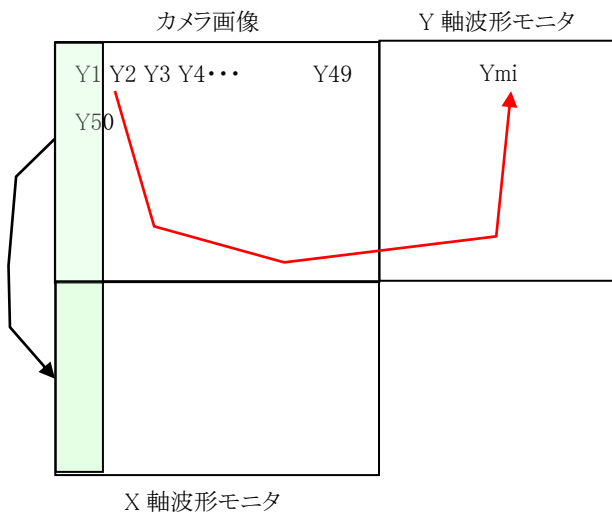
【図 9.4】 MAX 選択時のデータの取り扱い(例:輝度値)



※ $Y_{ma} = \text{Max}(Y1, Y2, Y3, \dots, Y49)$; 1Line 分の最大値

- MIN:カメラ画像の1LINE のデータを取得し、その最小値を波形モニタに表示します。

【図 9.5】 MIN 選択時のデータの取り扱い(例:輝度値)



※ $Y_{mi} = \text{Min}(Y1, Y2, Y3, \dots, Y49)$; 1Line 分の最小値

<Color>

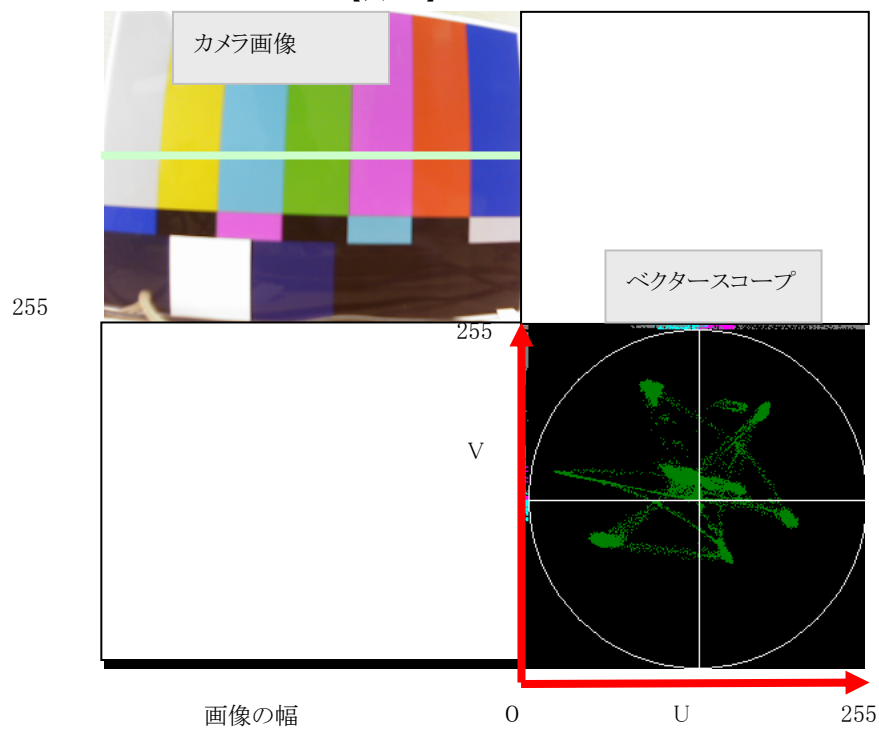
- Y:輝度情報を波形モニタに表示します。
- U:U 情報を波形モニタに表示します。
- V:V 輝度情報を波形モニタに表示します。

9.2. ベクタースコープ

ベクタースコープとは一般的に「色相の度合い」を表示するためのものになります。またカラーバー信号とあわせて利用することにより色相の調整を行うことができます。

本 SDK のベクタースコープは縦軸を V 情報振幅、横軸を U 情報振幅として表示します。

【図 9.6】 ベクタースコープ



<Mode>

- ALL:カメラ画像の全てのU・V値を取得し、その値をベクタースコープ上のU・V座標に反映します。

10. エラーメッセージ

モニタリング、レコーディング、その他処理中にエラーが発生することがあります。アプリケーションはエラーが発生した場合はメッセージボックスを表示し、ユーザーへエラー発生を通知します。エラー発生後、各処理に従い処理を終了します。以下の表にエラーメッセージ、エラー発生の要因、対処法を記載します。

10.1. カメラデバイスに関するエラー

(1) SVI ボード接続エラー

SVI ボードへの接続が行えない場合に発生します。

【図 10.1】 接続エラーメッセージ

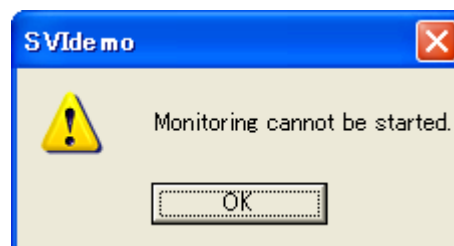


<p>要因</p> <p>SVI ボードの電源が入っていない、SVI ボードとホスト PC 間を USB ケーブル等で接続していない。</p>
<p>対処</p> <p>SVI ボードの電源の確認、SVI ボードとホスト PC 間の接続を確認してください。</p>

(2) モニタリング開始エラー

モニタリング開始時にステータスが異常となってモニタリングが開始できない場合に発生します。

【図 10.2】 モニタリング開始エラーメッセージ



<p>要因</p> <p>SVI ボード側の電源が入っていない、カメラの電源が入っていない、VHSync が動作していない等が考えられます。</p>
<p>対処</p> <p>SVI ボード及び、カメラの電源が入っていないか、またカメラの VHSync が動作しているか</p>

確認してください。

(3) モニタリングストップエラー

モニタリングを行っている際に、SVI ボードからフレームの取得ができなくなった場合にアプリケーションがエラーと認識して表示します。

【図 10.3】 モニタリングストップエラーメッセージ



要因

モニタリング動作中に SVI ボードとホスト PC 間の接続が切れた、SVI ボードの電源が切れた、または SVI ボードとカメラ間の接続が切れた可能性があります。

対処

SVI ボードの電源及び各接続を確認してください。

(4) レコーディング開始エラー

レコーディング開始時にステータスが異常となってレコーディングを開始できない場合に発生します。

【図 10.4】 レコーディング開始エラーメッセージ



要因

SVI ボード側の電源が入っていない、カメラの電源が入っていない、VHSync が動作していない等が考えられます。

対処

SVI ボード及び、カメラの電源が入っていないか、またカメラの VHSync が動作しているか確認してください。

(5) レコーディングタイムアウト

指定された秒数以上経過してもレコーディングが終了なく、かつレコーディングストップを

発行等を行って異常が発生した場合には表示します。

【図 10.5】レコーディングタイムアウトエラーメッセージ



要因

レコーディング中に SVI ボードの電源等が切れた可能性があります。

対処

SVI ボード及び、カメラの電源が入っていないか、またカメラの VHSync が動作しているか確認してください。

(6) ステータス取得エラー

SVI ボードからステータスを取得できない場合に表示されます。

【図 10.6】ステータス取得エラー



要因

SVI ボードとホスト PC 間の接続が確立できていない、SVI ボードの電源が切れている、または SVI ボードとカメラ間の接続が切れている可能性があります。

対処

SVI ボードの電源及び各接続を確認してください。

10.2. 設定条件に関するエラー

(1) 切り出し位置の奇数入力エラー

モニタリングの切り出し位置 (X) に対して奇数を設定した場合に表示されます。

【図 10.7】切り出し位置設定エラーメッセージ



要因

モニタリングの切り出し位置を設定する箇所に奇数を入力しているため

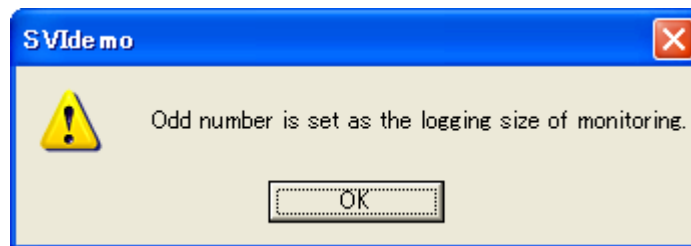
対処

偶数を入力してください。

(2) 切り出しサイズの奇数入力エラー

モニタリングの切り出しサイズ（横方向）に対して奇数を設定した場合に表示されます。

【図 10.8】 切り出しサイズ設定エラーメッセージ



要因

モニタリングの切り出しサイズを設定する箇所に奇数を入力しているため

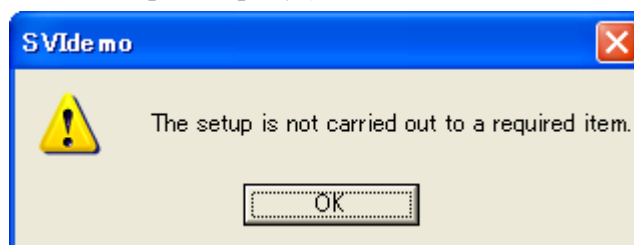
対処

偶数を入力してください。

(3) 未設定エラー

各動作で必須の値が設定されていない場合に表示されます。

【図 10.9】 未設定エラーメッセージ



要因

必要な項目へ値の入力等設定がされていないため

対処

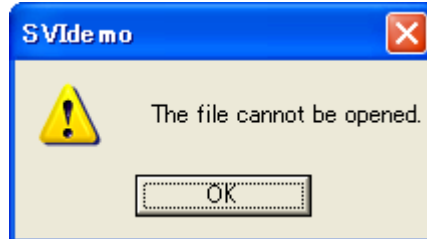
必要項目へ値を設定してください

10.3. ファイルアクセスに関するエラー

(1) ファイルのオープンエラー

ファイルを開けない場合に発生するエラーです。

【図 10.10】 ファイルオープンエラーメッセージ



要因

指定されたファイルがオープンできない。既に別のアプリケーションでオープンされている、または指定したパスにファイルが見つからない等があります。

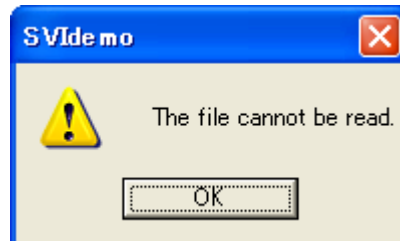
対処

別のアプリケーションが開いているのであれば、そのアプリケーションを閉じて下さい。指定したパスにファイルが存在しない場合は正しいパスを設定してください。

(2) ファイルの読み込みエラー

ファイルを読み込めない場合に発生するエラーです。

【図 10.11】 ファイル読み込みエラーメッセージ



要因

空ファイル等である、または読み込みロックがかかっている可能性があります。

対処

ファイルサイズを確認してください。また別のアプリケーションでファイルをオープンしている可能性もあるので、別のアプリケーションで使用していないか確認してください。

(3) ファイルの書き込みエラー

ファイルへ書き込めない場合に発生するエラーです。

【図 10.12】 ファイル書き込みエラーメッセージ

**要因**

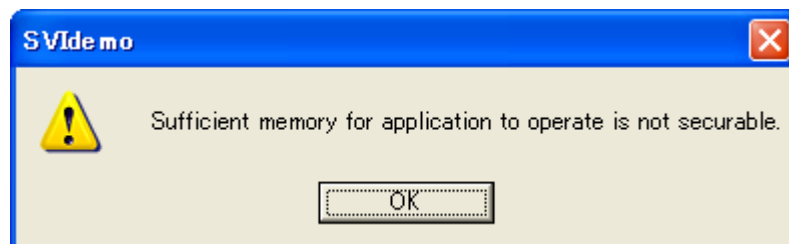
ファイルが読み取り専用である、または既に別ファイルで開いているためロックがかかっている可能性があります。

対処

ファイル情報を確認してください。別のアプリケーションで使用していないか確認してください。

10.4. アプリケーションの動作に関するエラー**(1) メモリ動的確保失敗エラー**

メモリの動的確保に失敗した場合に表示されます。

【図 10.13】メモリ動的確保エラーメッセージ**要因**

画像サイズが大きい、OS の管理している空きメモリが少ない場合に発生する可能性があります。

対処

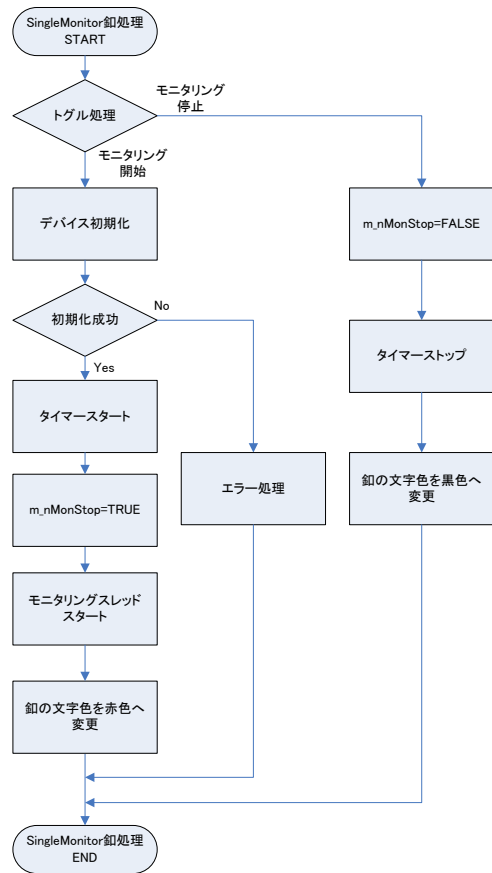
他のアプリケーションを終了して OS に十分な空きメモリを与えてください。

11. 各種動作フロー

シングルモニタリング、マルチモニタリング、モニタリングスレッド、レコーディング及びインフォメーションの動作フローならびに、SDK でのシングルモニタリングとモニタリングスレッドの正常動作時のシーケンス図を示します。なお SVI API コールルーチンのフローについては別章に記載してありますのでそちらをご覧ください。

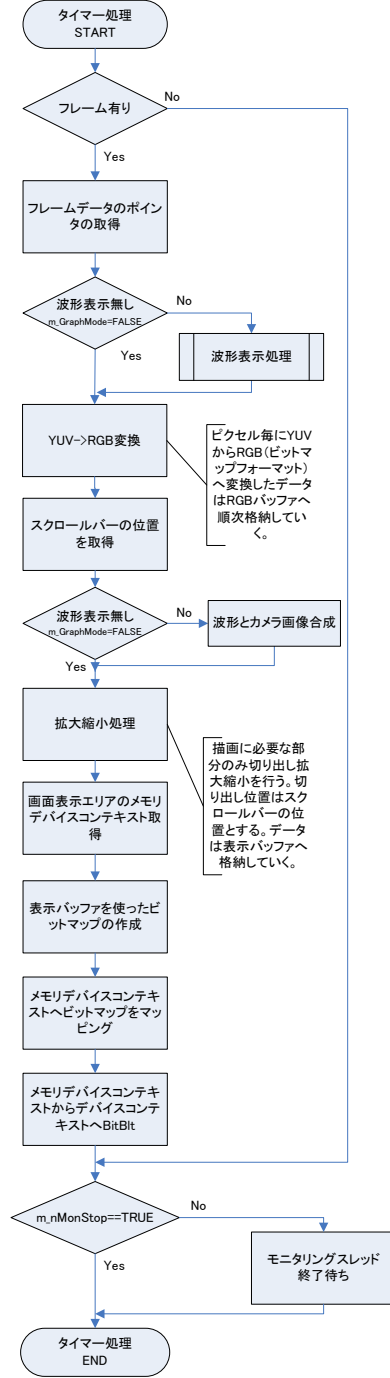
11.1. シングルモニタリング動作フロー

SingleMonitorボタンフロー

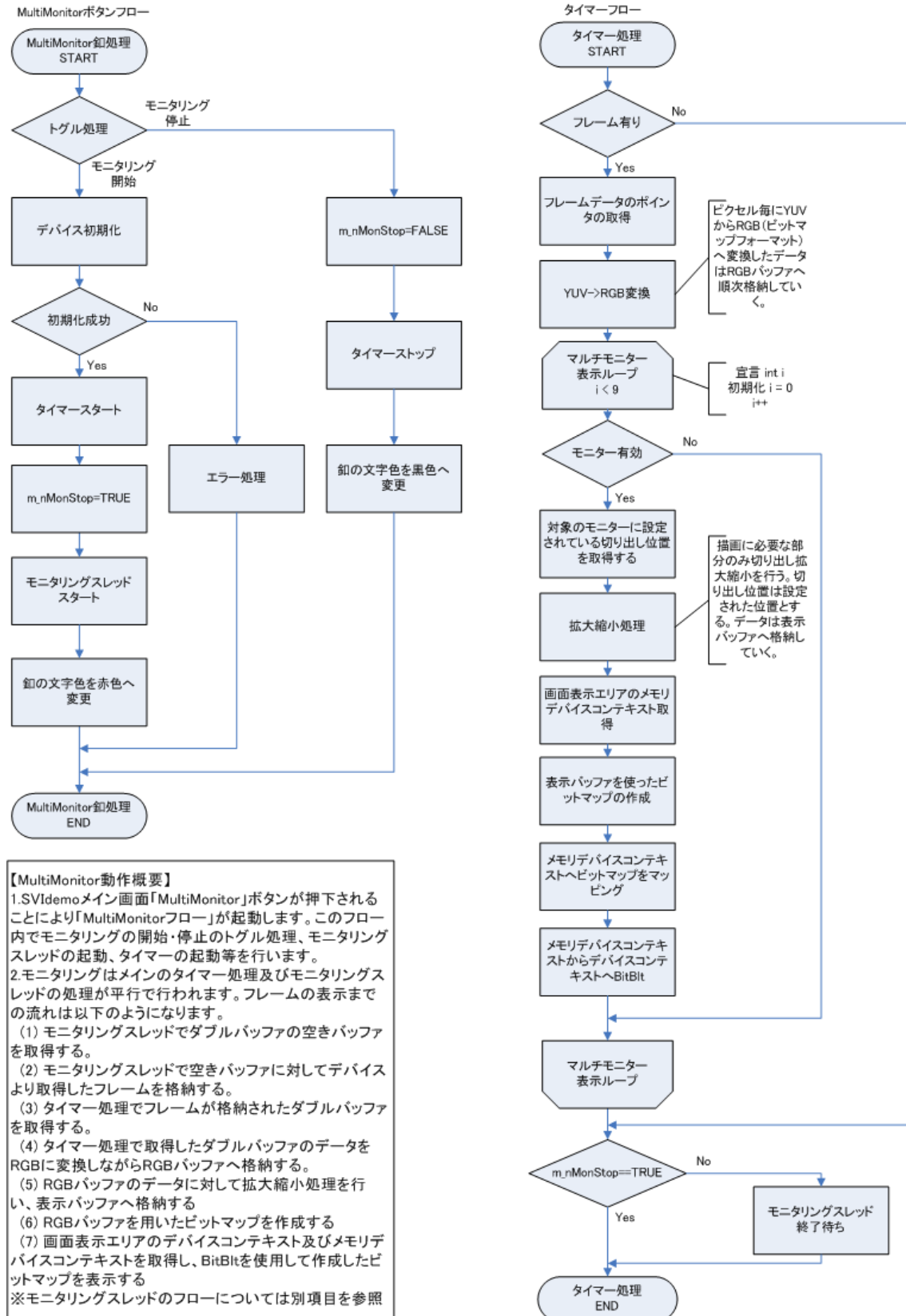


【SingleMonitor動作概要】
 1. SVIdemoメイン画面「SingleMonitor」ボタンが押下されることにより「SingleMonitorフロー」が起動します。このフロー内でモニタリングの開始・停止のトグル処理、モニタリングスレッドの起動、タイマーの起動等を行います。
 2. モニタリングはメインのタイマー処理及びモニタリングスレッドの処理が平行で行われます。フレームの表示までの流れは以下のようになります。
 (1) モニタリングスレッドでダブルバッファの空きバッファを取得する。
 (2) モニタリングスレッドで空きバッファに対してデバイスより取得したフレームを格納する。
 (3) タイマー処理でフレームが格納されたダブルバッファを取得する。
 (4) タイマー処理で取得したダブルバッファのデータをRGBに変換しながらRGBバッファへ格納する。
 (5) 波形表示を行う場合には波形表示処理を行いRGBバッファに格納する。
 (6) RGBバッファのデータに対して拡大縮小処理を行い、表示バッファへ格納する
 (7) RGBバッファを用いたビットマップを作成する
 (8) 画面表示エリアのデバイスコンテキスト及びメモリデバイスコンテキストを取得し、BitBltを使用して作成したビットマップを表示する
 ※モニタリングスレッドのフロー、グラフ表示については別項目を参照

タイマーフロー

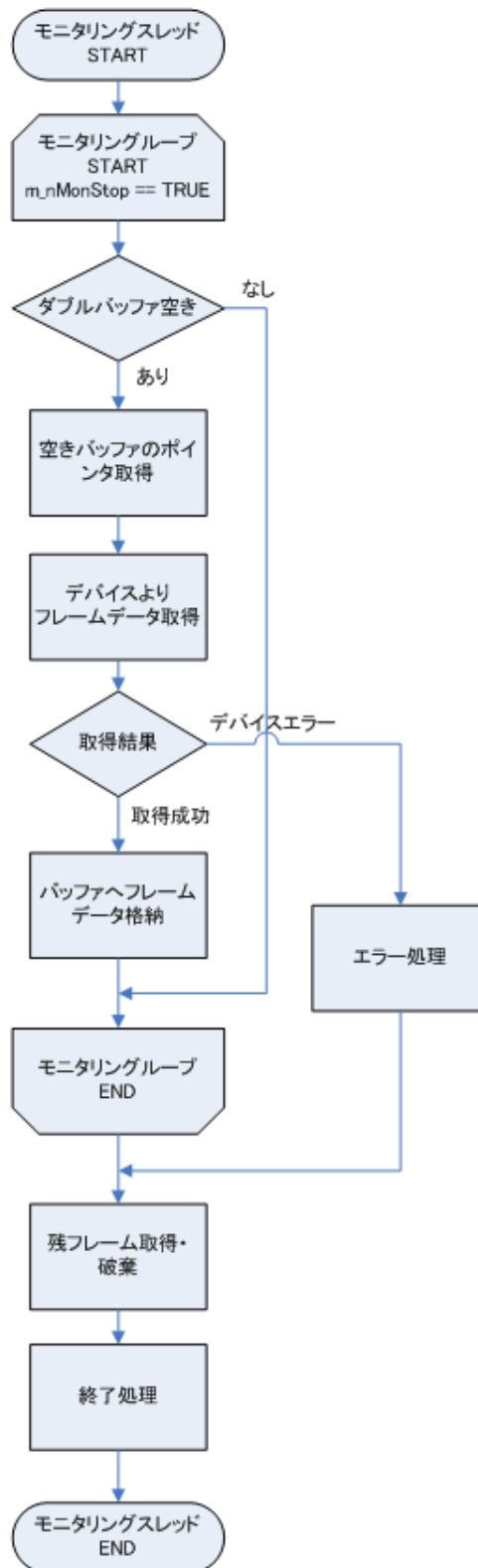


11.2. マルチモニタリング動作フロー

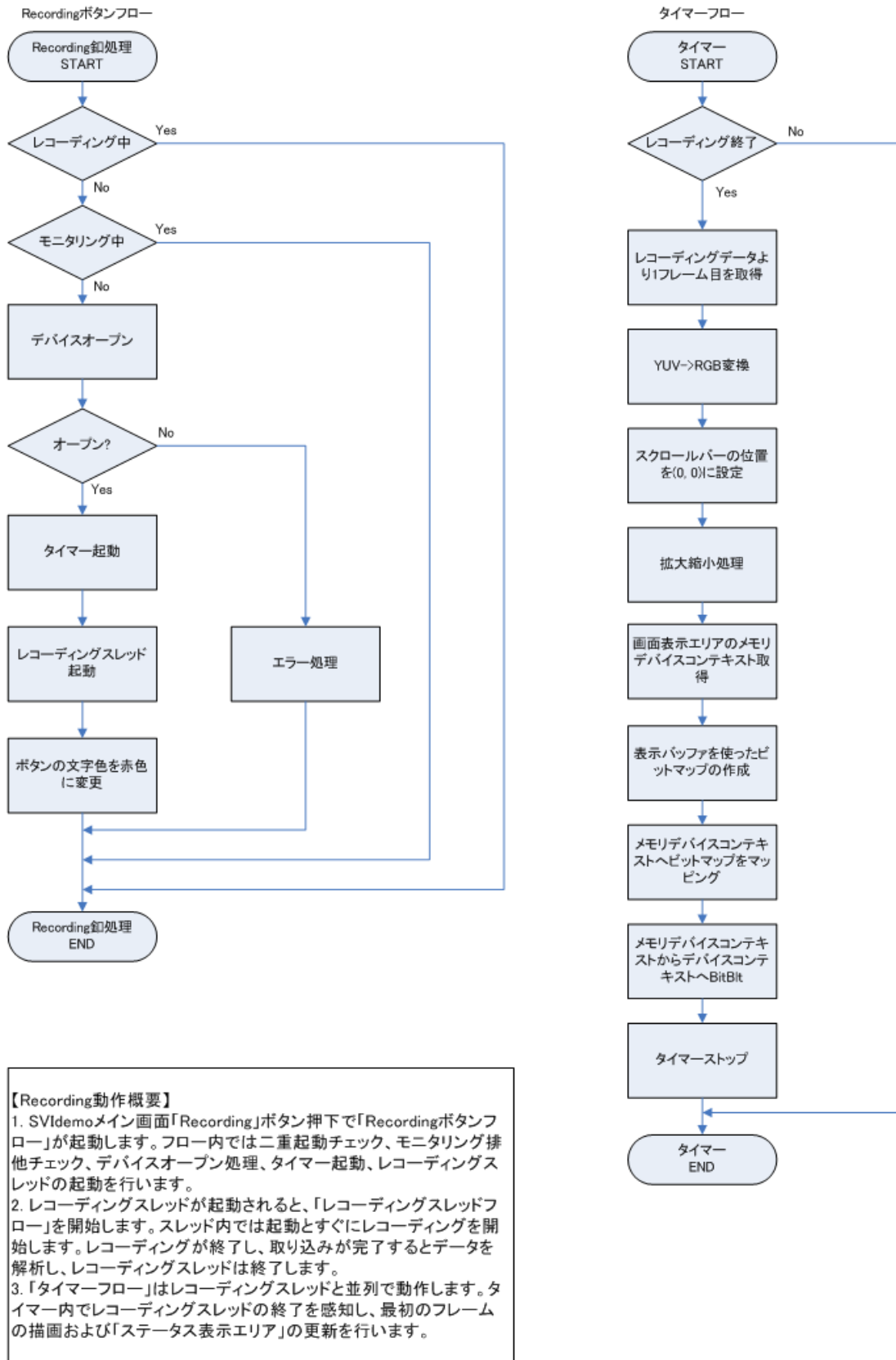


11.3. モニタリングスレッドフロー

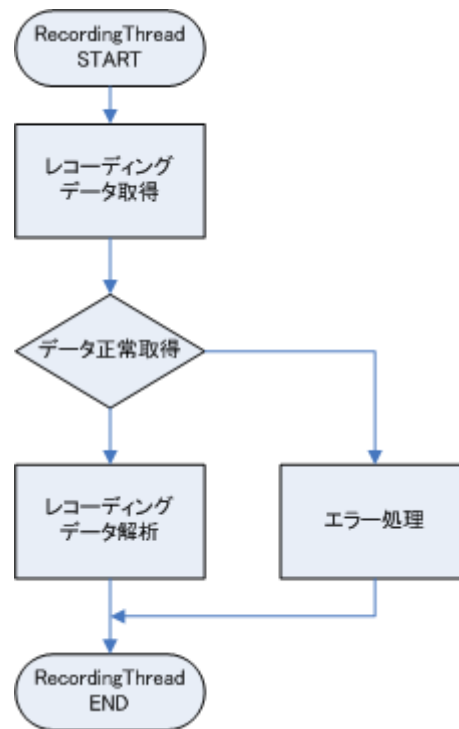
モニタリングスレッドフロー



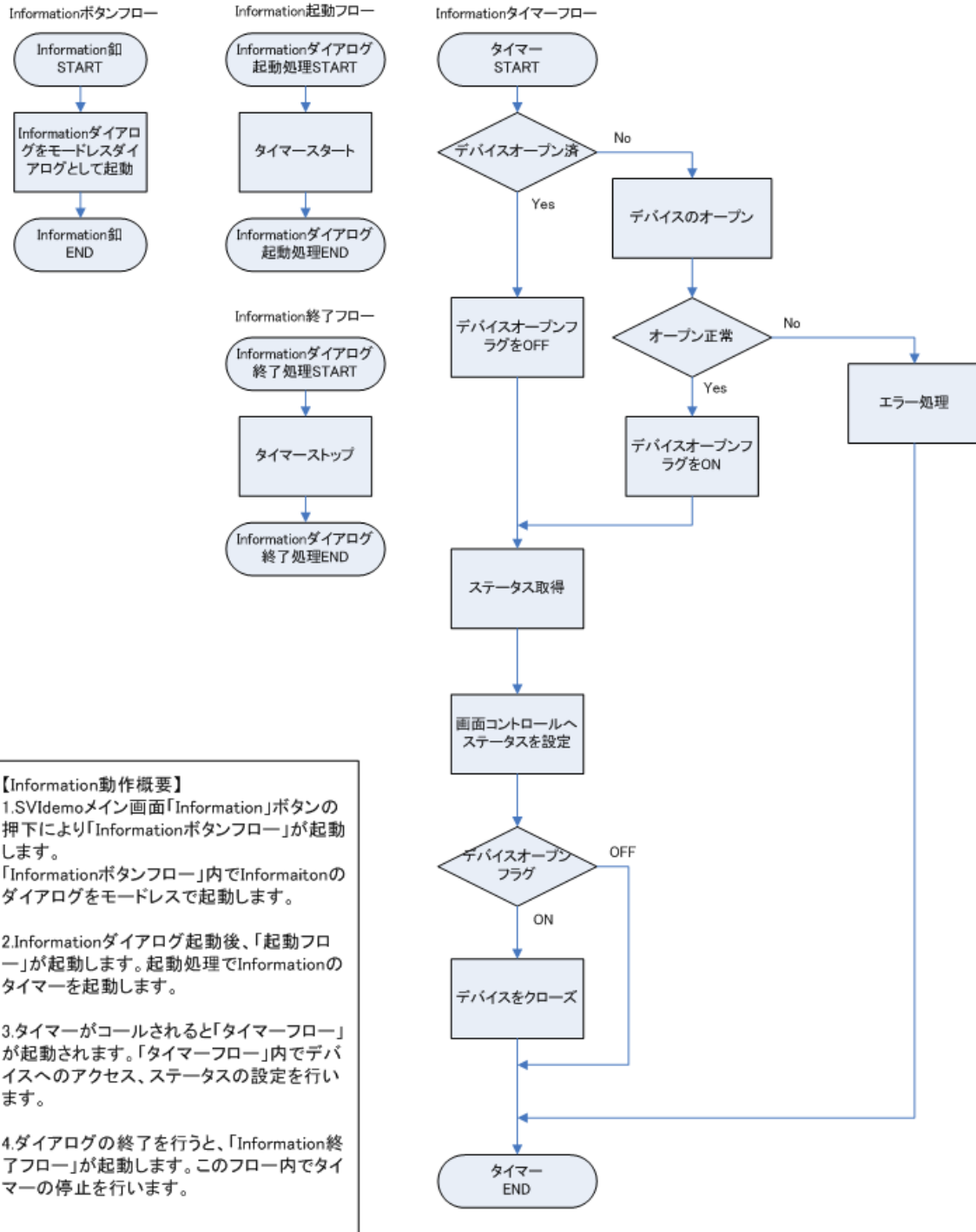
11.4. レコーディング動作フロー



11.5. レコーディングスレッドフロー

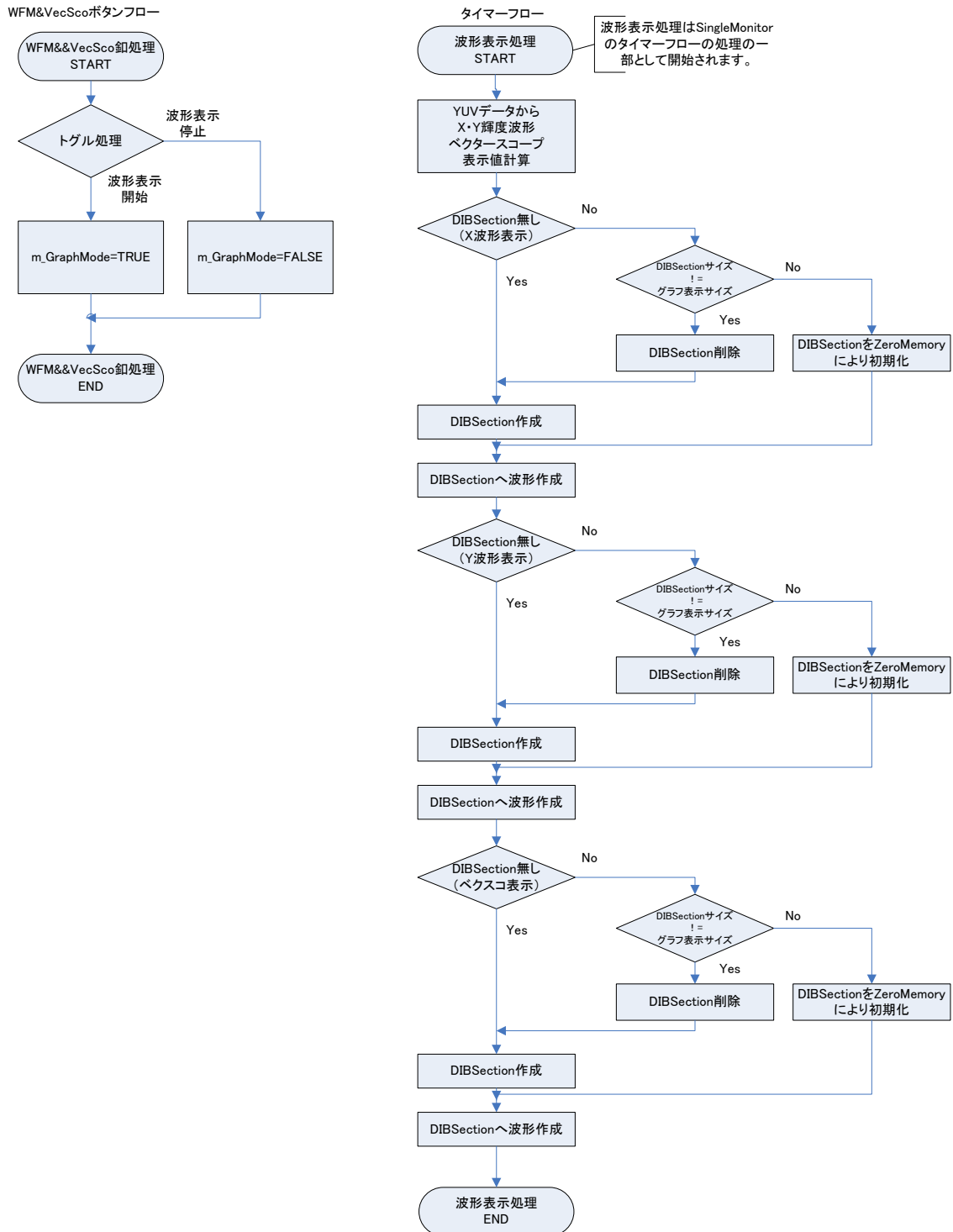


11.6. インフォメーション動作フロー

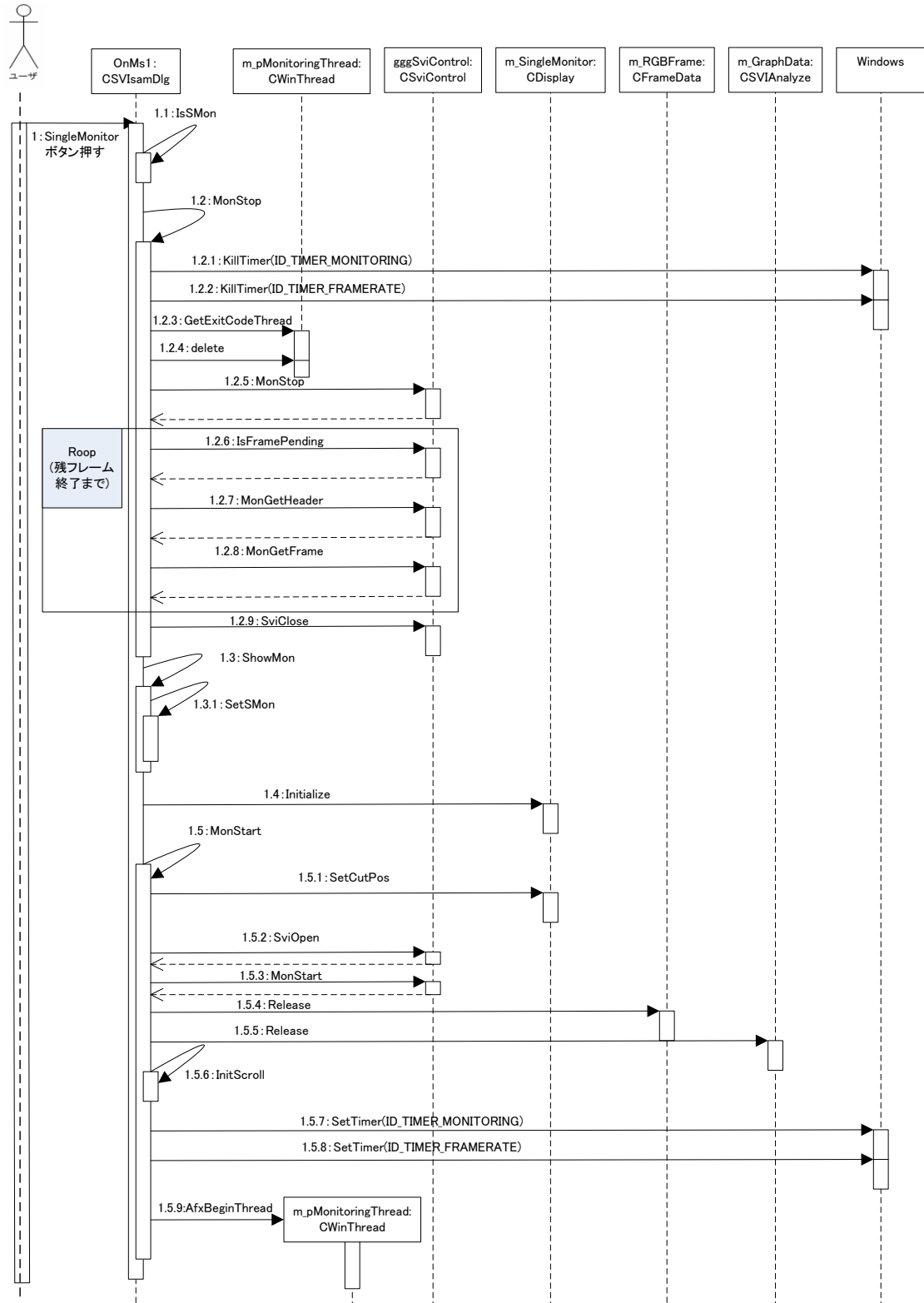


11.7. 波形表示動作フロー

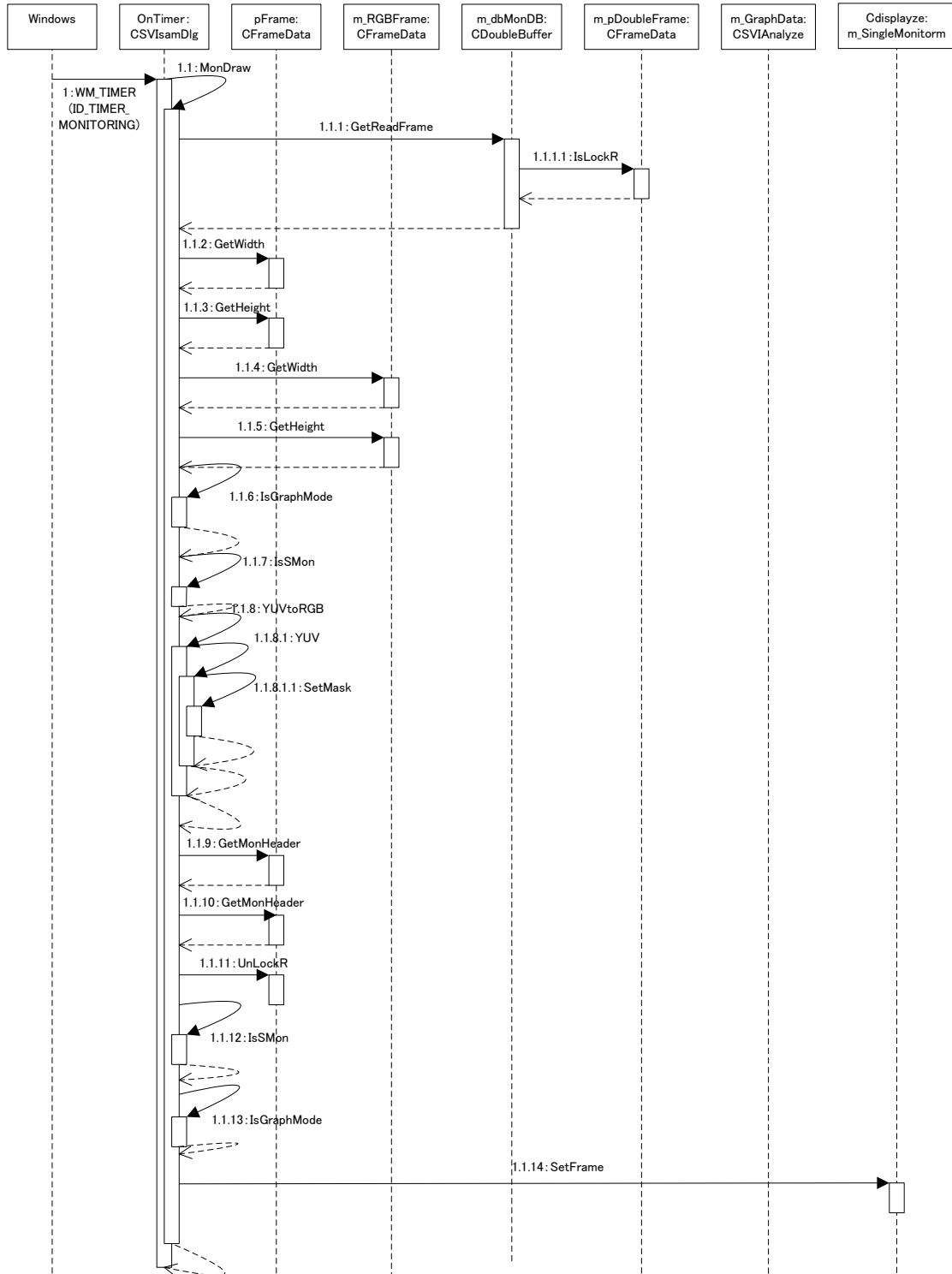
波形表示タイマーフローについてはシングルモニター処理のタイマーフロー内「波形表示処理」の詳細を示したものである



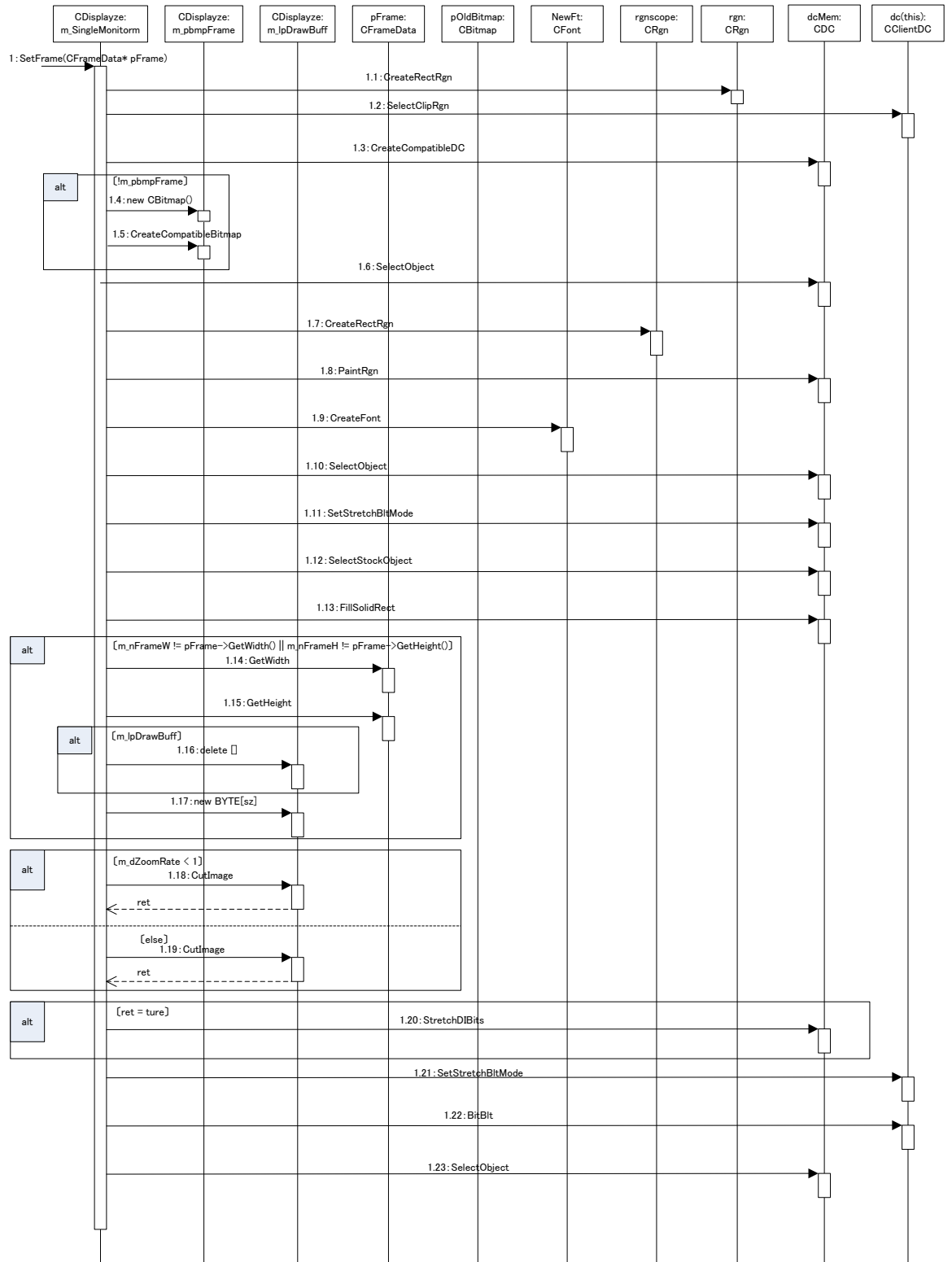
11.8. シングルモニター正常動作シーケンス



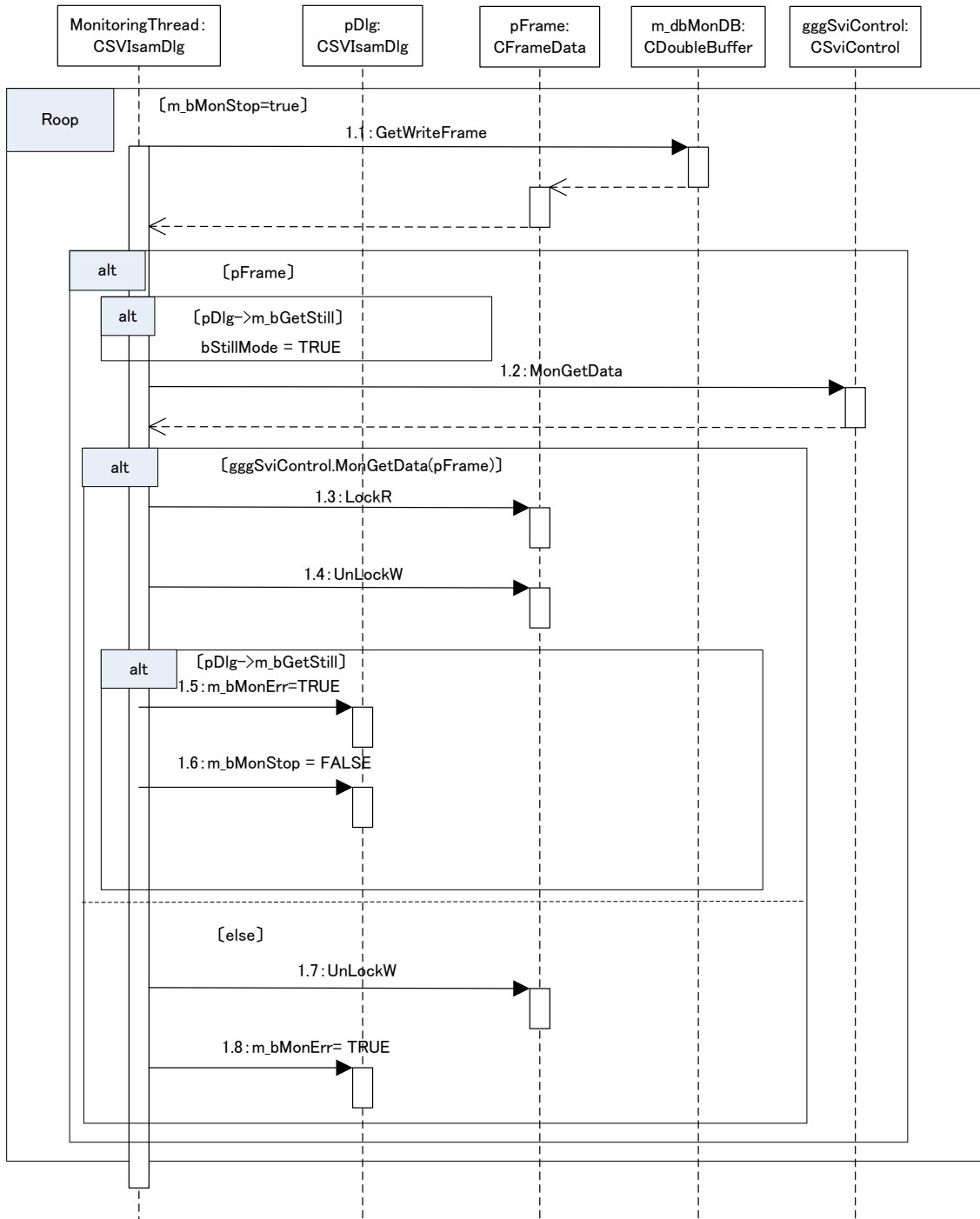
11.9. シングルモニターOntimer 正常動作シーケンス(モニタリング・グラフ表示なしの処理)



11.10. シングルモニターSetFrame 正常動作シーケンス(モニタリング・グラフ表示なしの処理)



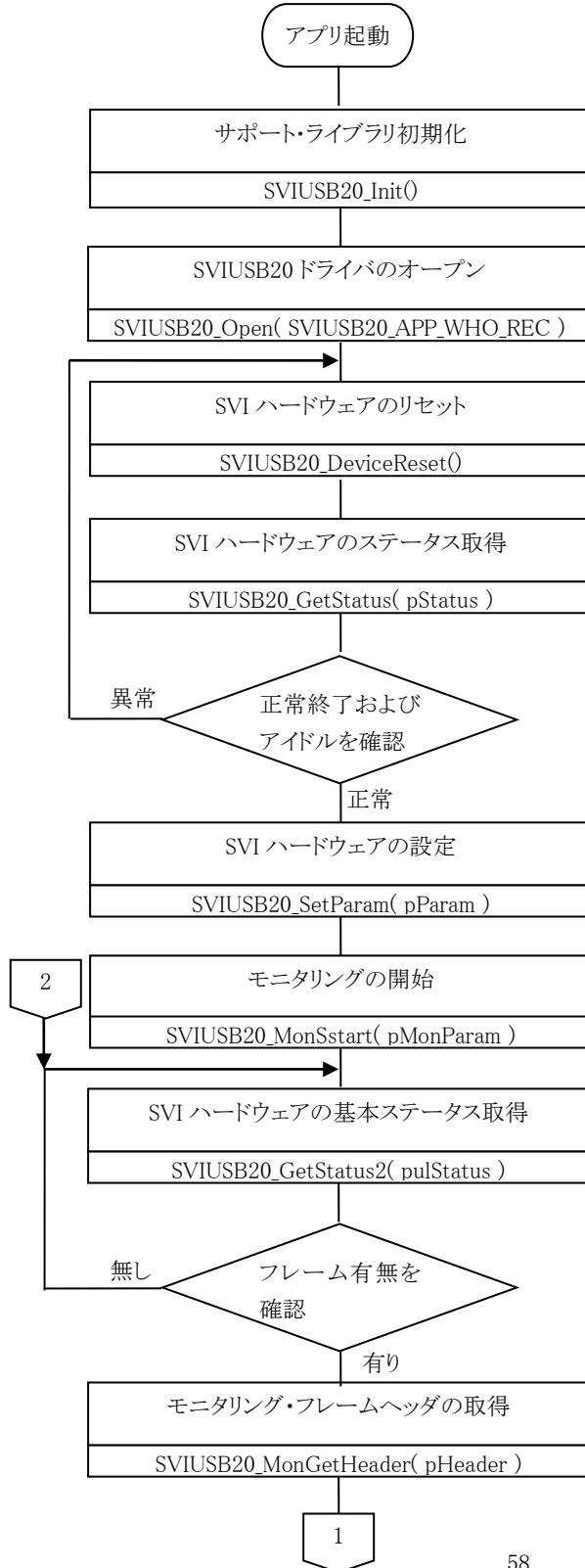
11.11. モニタリングスレッド正常動作シーケンス



12. SVI API 各種フロー

各動作での API 使用例を以下に記述します。この使用例は SVI 画像入カライブラリ説明書より抜粋してあります。

12.1. モニタリングモード時の画像入カライブラリ使用例



※各 API コール後エラー処理を行って下さい。

※必ず最初に行ってください。

※2 重オープンはできないので気を付けて下さい。

※必ず行って下さい。また調子が悪い時も有効です。

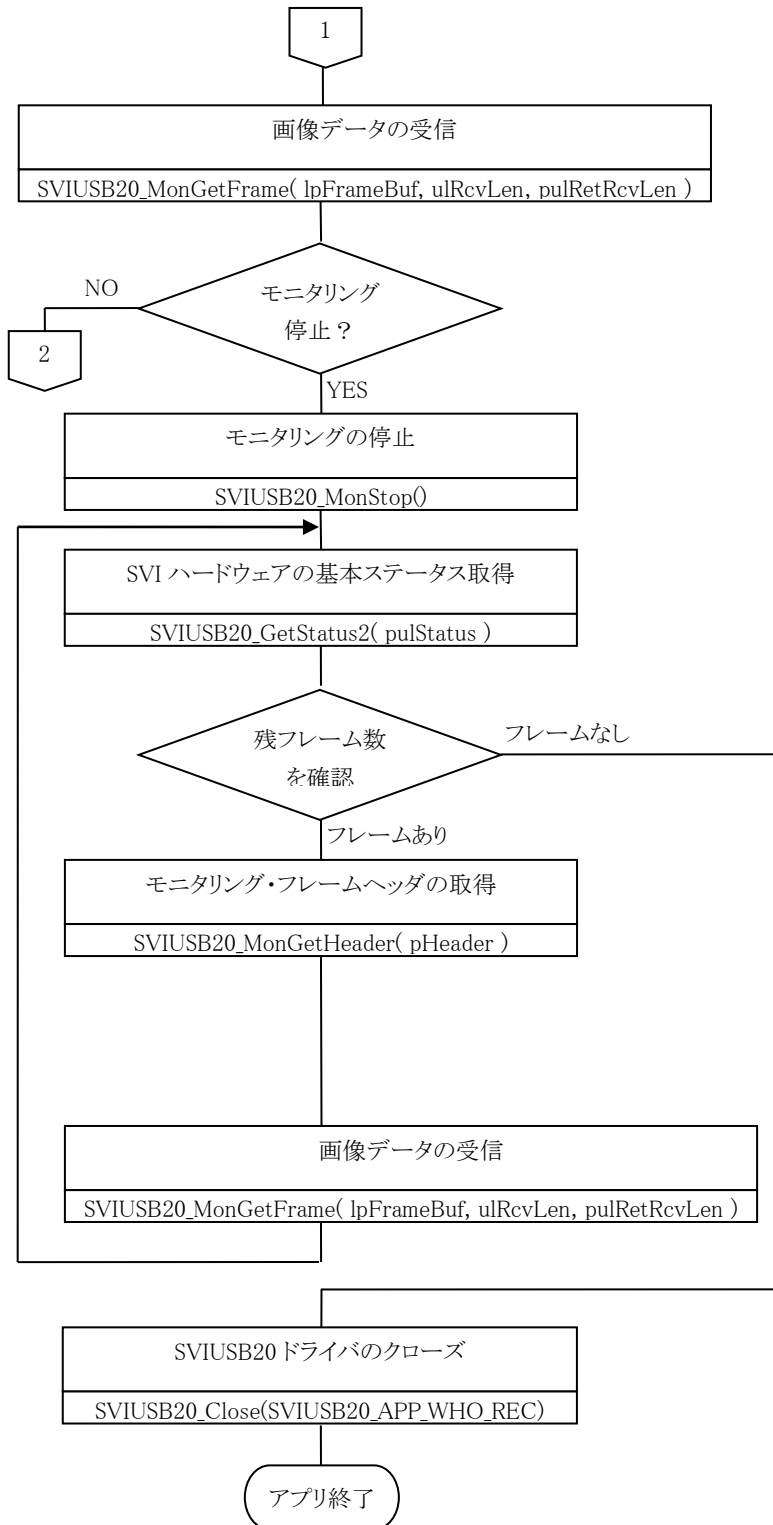
※基本ステータスだけであれば SVIUSB20_GetStatus2(pulStatus)でも可能です。

※3 回リトライしても異常であればエラー終了の手続きをして下さい。

※必ず行って下さい。

※ SVIUSB20_MonSstart コール後、SVIUSB20_GetStatus2 で、SVI ハードウェアにキャプチャ画像フレームの有無を確認します。

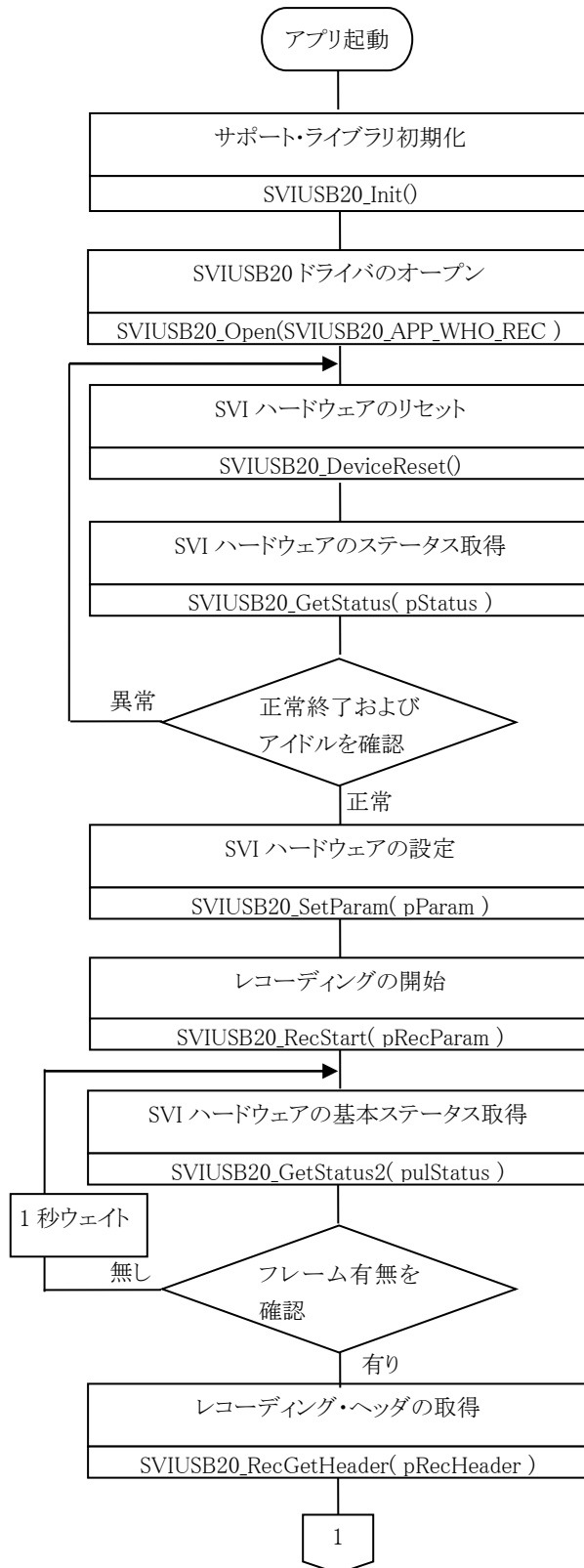
※取得したフレームヘッダを参照して ulRcvDataSize を求めて下さい。



※モニタリング停止でなければ連続して受信して下さい。

※停止後、基本ステータスを取得し、残フレームの有無を確認します。残フレームがなくなるまで受信動作を繰り返します。

12.2. レコーディングモード時の画像入カライブラリ使用例



※各 API コール後エラー処理を行って下さい。

※必ず最初に行ってください。

※2 重オープンはできないので気を付けて下さい。

※必ず行って下さい。また調子が悪い時も有効です。

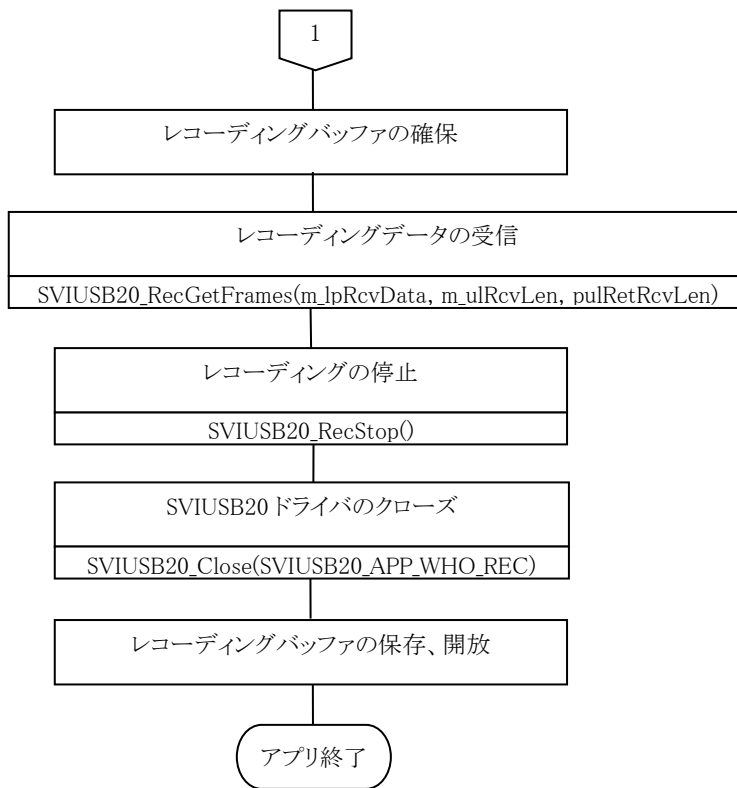
※基本ステータスであれば SVIUSB20_GetStatus2(pulStatus)でも可能です。

※3 回リトライしても異常であればエラー終了の手続きをして下さい。

※必ず行って下さい。

※ SVIUSB20_RecStart コール後、SVIUSB20_GetStatus2 で、SVI ハードウェアにキャプチャ画像フレームの有無を確認します。

※取得したヘッダから RecHeader.ulNumScan がレコーディングした実バイト数です。この値を 4 で割り切れるよう桁上げた値が転送バイト数になります。



4の倍数に切り上げた値でメモリを確保してください。

m_lpRcvData は予め m_ulRcvLen 分確保しておいて下さい。

12.3. SPIによるコマンド送信時の画像入カライブラリ使用例



※各 API コール後エラー処理を行って下さい。

※必ず最初に行ってください。

※2 重オープンはできないので気を付けて下さい。

※チップ ID、スタートアドレスなど指定します。

12.4. SPI によるコマンド受信時の画像入力ライブラリ使用例



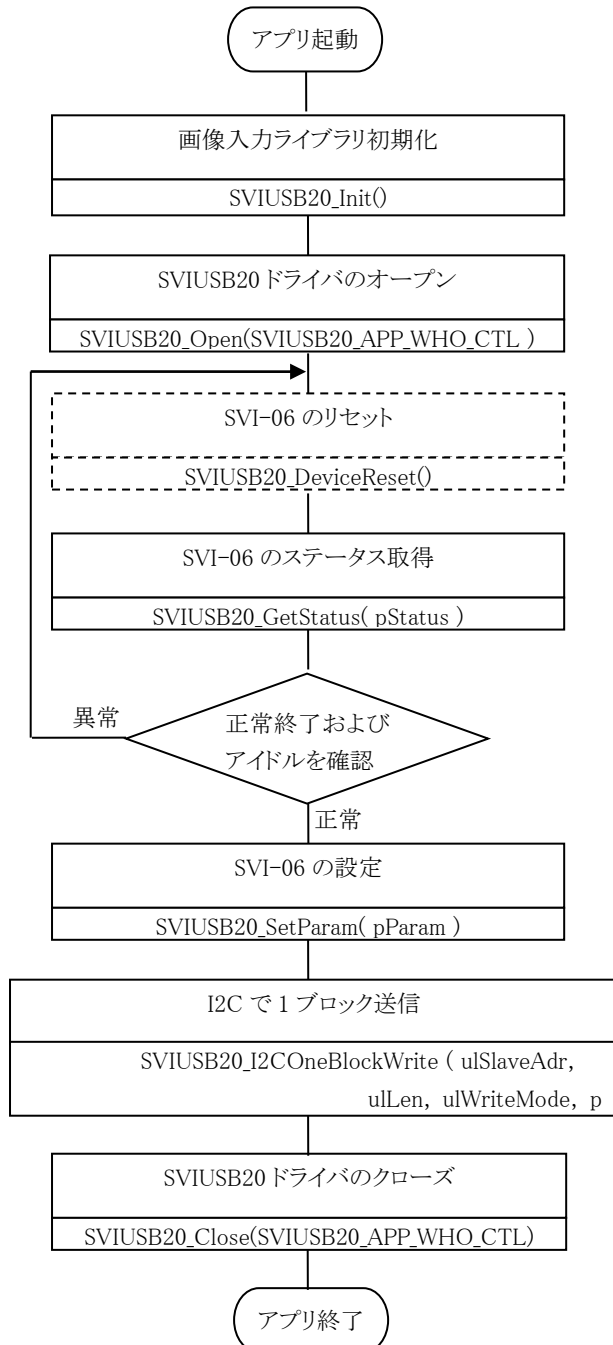
※各 API コール後エラー処理を行って下さい。

※必ず最初に行ってください。

※2 重オープンはできないので気を付けて下さい。

※チップ ID、スタートアドレスなど指定します。

12.5. I2C によるコマンド送信時の画像入カライブラリ使用例



※各 API コール後エラー処理を行って下さい。

※必ず最初に行ってください。

※2 重オープンはできないので気を付けて下さい。

※必要あれば行って下さい。また調子が悪い時にも有効です。(3 秒かかります)

※基本ステータスだけであれば SVIUSB20_GetStatus2(pulStatus)でも可能です。

※3 回リトライしても異常であればエラー終了の手続きをして下さい。

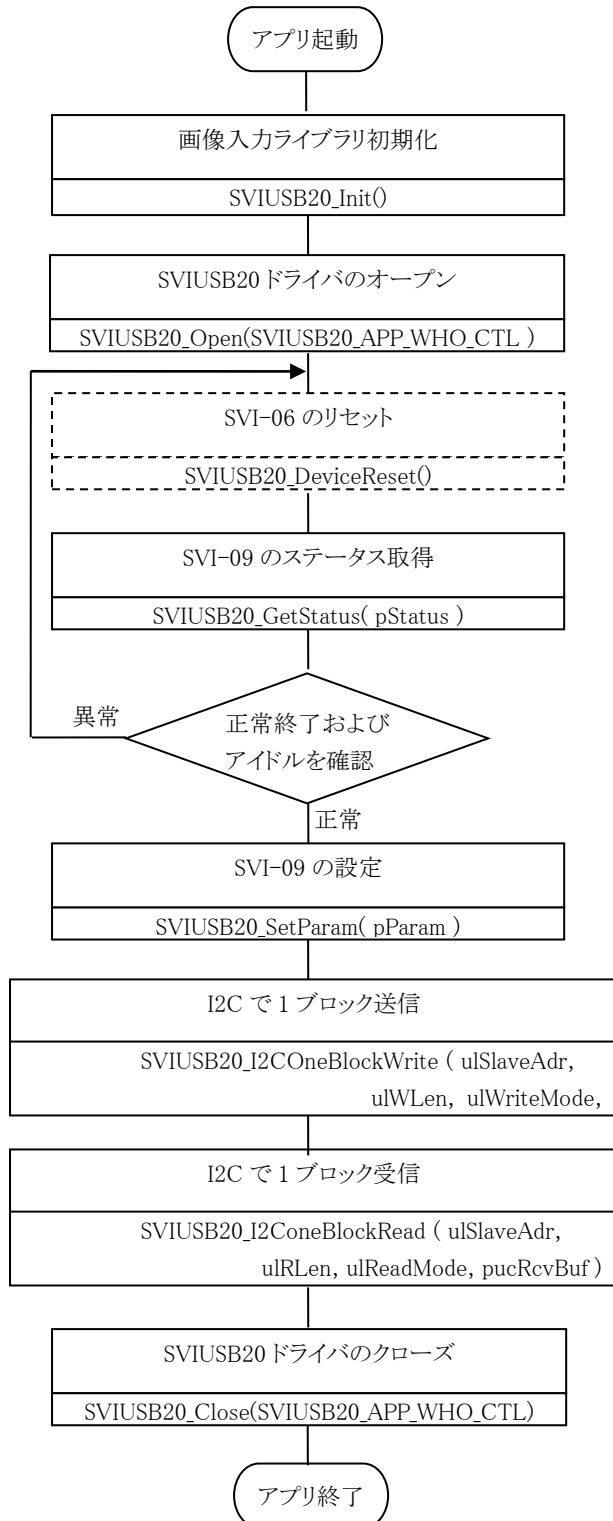
※必ず行って下さい。

※ulSlaveAdr にはスレーブ ID を代入する。

(API の中で左に 1 ビットシフトしている)

※ulLen にはスレーブ ID を含まないサブアドレスからのバイト数を指定する。

12.6. I2C によるコマンド受信時の画像入カライブラリ使用例



※各 API コール後エラー処理を行って下さい。

※必ず最初に行ってください。

※2 重オープンはできないので気を付けて下さい。

※必要あれば行って下さい。また調子が悪い時も有効です。(3 秒かかります)

※基本ステータスだけであれば SVIUSB20_GetStatus2(pulStatus)でも可能です。

※3 回リトライしても異常であればエラー終了の手続きをして下さい。

※必ず行って下さい。

※ulSlaveAdr にはスレーブ ID を代入する。

(API の中で左に 1 ビットシフトしている)

※ulWLen にはサブアドレスのみなので1を代入する。

※ulRLen には受信するバイト数を指定する。

12.7. 複数台の SVI-09 を接続する場合の API 使用例

1 台の PC に複数台の SVI-09 を接続する場合、SVIUSB20_EnumDevice API で接続している SVI-09 ボードの論理番号と DSW 設定番号を取得することができます。

まず、SVIUSB20_EnumDevice API を使用して、何台の SVI-09 が接続されているかを取得します。次に接続されている数分の情報エリアを確保した後、再度 SVIUSB20_EnumDevice API を呼び出し、接続されている SVI-09 の情報を取得します。INI ファイルに“BOARDNO”というキーワードでボード番号を記述しますので、この番号と SVIUSB20_EnumDevice API で取得した情報内のボード番号が一致する SVI-09 を見つけます。見つけたら SVIUSB20_DeviceSelect API で接続をします。以降は前項の操作を行って下さい。

SVI-09 の使用を終了した後は、SVIUSB20_DeviceRelease API で SVI-09 を開放する必要があります。忘れずに行ってください。

これらの操作を行っているソースファイル及び箇所は、SviControl.cpp の SviOpen 関数と SviClose 関数です。一度ご覧になってください。

12.8. 複数台の SVI-09 を使用する場合の使用環境について

1 台の PC に複数の SVI-09 を接続して使用する場合は、フォルダを分けて、別プロセスとして 2 つの SVIsam.exe を起動してご使用下さい。

CD-ROM 内の SVIsam¥bin-0 と SVIsam¥bin-1 を HDD のフォルダにコピーして下さい。各 bin-x フォルダの svisdk.ini の“BOARDNO”キーワードに続く番号でボード番号を指定しています。SVI-09 ボードに“ボード No. 0”または“ボード No. 1”のシールが貼ってありますので、bin-0 での SVIsam.exe はボード番号 0 番の SVI-09 ボードを接続、bin-1 での SVIsam.exe はボード番号 1 番の SVI-09 ボードを接続して使用します。

13. ソースファイル一覧

この章では SVIsam アプリケーションを構成する各ファイルの概要を一覧表で示します。なお開発ツールで自動的に作成されるファイルに関しては割愛させて頂いている箇所もあります。

ファイル名	概要
Comm.cpp、Comm.h	共通関数を格納
Display.cpp、Display.h	画像表示関数を格納
DoubleBuffer.cpp、DoubleBuffer.h	モニタリング時のダブルバッファ制御関数を格納
FrameData.cpp、FrameData.h	フレームデータ制御の関数を格納
ParamList.cpp、ParamList.h	パラメータファイルオブジェクト関数を格納
ParamWrite.cpp、ParamWrite.h	パラメータファイルオブジェクトへの追加関数を格納
RecData.cpp、RecData.h	レコーディングデータアクセス関数を格納
RecLongDlg.cpp、RecLongDlg.h	長時間録画関数を格納
RecLongSettingDlg.cpp RecLongSettingDlg.h	長時間録画設定ダイアログ関連を格納
SVIANalyze.cpp、SVIANalyze.h	ウェブフォーム、ベクタースコープ処理関数を格納
SVIcbDlg.cpp、SVIcbDlg.h	カラーバーダイアログ関連を格納
SVIcmdsetDlg.cpp、SVIcmdsetDlg.h	Setting ダイアログ関連を格納
SviControl.cpp、SviControl.h	SVI ボードアクセス関数を格納
SVIsam.cpp、SVIsam.h	SVIsam メイン関数を格納
SVIsamDlg.cpp、SVIsamDlg.h	SVIsam ダイアログ関連を格納(実質メイン)
SVIWFInitDlg.cpp、SVIWFInitDlg.h	ウェブフォーム、ベクタースコープダイアログ関連を格納
KizuDlg.cpp、KizuDlg.h	黒キズ検査ダイアログ関連を格納
IniControl.h	INI ファイルアクセス関数を格納
SVIUSB20.h	SVI 画像入カライブラリのヘッダー
SVIUSB20.lib	SVI 画像入カライブラリのインポートライブラリ
SVIsam.dsw	SVIsam プロジェクトワークスペース

14. 動作環境

PC 本体	: Windows10 が問題なく動作すること
CPU	: Intel CORE i5 1.4GHz 以上推奨
メモリ	: 8GB 以上推奨
HDD	: SSD500GB 以上推奨
USB	: USB3.0 インターフェースを搭載するもの
OS	: Windows10

15. 開発環境

コンパイラ/リンカ	: Microsoft VisualStudio 2008 (SP1)
-----------	-------------------------------------