



# カメラライブラリ マニュアル

このマニュアルは、カメラライブラリの仕様  
について記載します。

#### **ご注意**

- このソフトウェアおよびマニュアルの、一部または全部を無断で使用、複製することはできません。
- このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。
- このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。
- このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なしに変更することがあります。
- このマニュアルの著作権はカシオ計算機株式会社に帰属します。
- 本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2018 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。

## 変更履歴

バージョン	変更日付	ページ	内容
1.00	2008.11		新規作成
1.01	2009.03	2	開発環境とプログラミング言語の対応表を訂正
		20	CAMGetLightMode 関数の戻り値を追加
		24	CAMGetDigitalZoom 関数の戻り値を追加
		28	CAMSingleCaptureEx 関数の解説の誤記を訂正
		34	CAMSetIllumination 関数の宣言の誤記を訂正
		35	CAMGetIllumination 関数の宣言の誤記を訂正
		37	CAMGetIris 関数の宣言の誤記を訂正
		45	CAMSetFlash 関数の宣言の誤記を訂正
		47	CAMSetFocus 関数の宣言の誤記を訂正
		52	CAMSetPreviewMode 関数の宣言の誤記を訂正
1.02	2009.07	-	対象機種に DT-5300 を追加
		12-14	CAMSingleCapture 関数の注意を追加
		28-29	CAMSingleCaptureEx 関数の注意を追加
		33	CAMIlluminationOn 関数の注意を追加
		54-55	CAMSetVideoCaptureSettings 関数を追加
		56-57	CAMGetVideoCaptureSettings 関数を追加
		58-59	CAMSetPreviewFrameRate 関数を追加
		60-61	CAMGetPreviewFrameRate 関数を追加
		62-63	CAMSetCaptureMode 関数を追加
		64	CAMGetCaptureMode 関数を追加
		65-66	CAMStartVideoCapture 関数を追加
		68-69	CAMStopVideoCapture 関数を追加
		70-71	CAMWaitForVideoEncode 関数を追加
		72	CAMCancelVideoEncode 関数を追加
1.03	2009.09	65	CAMStartVideoCapture 関数に注意追加
		68	CAMStopVideoCapture 関数に解説追加
		70	CAMWaitForVideoEncode 関数のパラメータを修正 および注意追加
1.04	2009.12	27	CAMSingleCaptureEx 関数に解説追加
1.05	2011.07	8	CAMStartPreview 関数に解説追加
		12	CAMSingleCapture 関数に解説追加
		21	CAMSetDigitalZoom 関数に解説追加
		28	CAMSingleCaptureEx 関数に解説追加
1.06	2014.11	-	対象機種に IT-G500 を追加
		-	関数一覧表をデバイスライブラリ基本マニュアルへ移動
		4	CAM_IMAGE_SET 構造体に機種別対応表を追加
		8	CAMStartPreview 関数にパラメータおよび機種別対応 表を追加
		12	CAMSingleCapture 関数にパラメータおよび機種別対応 表を追加
		21	CAMSetDigitalZoom 関数に機種別対応表を追加
		28	CAMSingleCaptureEx 関数にパラメータおよび機種別対 応表を追加

[illegible]

# 目次

1. 概要	1
2. 動作環境	2
3. 構造体	4
3.1 CAM_IMAGE_SET	4
4. 関数	6
4.1 CAMOpen	6
4.2 CAMClose	7
4.3 CAMStartPreview	8
4.4 CAMStopPreview	11
4.5 CAMSingleCapture	12
4.6 CAMSetImageSettings	15
4.7 CAMGetImageSettings	16
4.8 CAMContinuousCapture	17
4.9 CAMSetLightMode	19
4.10 CAMGetLightMode	20
4.11 CAMSetDigitalZoom	21
4.12 CAMGetDigitalZoom	24
4.13 CAMSetSensitivity	25
4.14 CAMGetSensitivity	26
4.15 CAMRedrawPreview	27
4.16 CAMSingleCaptureEx	28
4.17 CAMIlluminationOn	33
4.18 CAMSetIllumination	34
4.19 CAMGetIllumination	35
4.20 CAMSetIris	36
4.21 CAMGetIris	37
4.22 CAMSetFlickerless	38
4.23 CAMGetFlickerless	40
4.24 CAMSetDigitalFlashMode	41
4.25 CAMGetDigitalFlashMode	43
4.26 CAMSetFlash	45
4.27 CAMGetFlash	46
4.28 CAMSetFocus	47
4.29 CAMGetFocus	49
4.30 CAMGetFocusStatus	50
4.31 CAMAutoFocus	51
4.32 CAMSetPreviewMode	52
4.33 CAMGetPreviewMode	53
4.34 CAMSetVideoCaptureSettings	54
4.35 CAMGetVideoCaptureSettings	56
4.36 CAMSetPreviewFrameRate	58
4.37 CAMGetPreviewFrameRate	60
4.38 CAMSetCaptureMode	62
4.39 CAMGetCaptureMode	64
4.40 CAMStartVideoCapture	65

4.41	CAMStopVideoCapture	68
4.42	CAMWaitForVideoEncode	70
4.43	CAMCancelVideoEncode	72
5.	プログラミング上の注意点	73
5.1	動作フローについて	73
6.	Device Emulator	75
6.1	CameraLib.inf	75
6.2	***.bmp	76
6.3	***.jpg	76

## 1. 概要

カメラライブラリは、本体に内蔵しているカメラモジュールを利用して、撮影機能、および撮影画像のファイルへの変換機能を提供します。

カメラクラスライブラリは、カメラライブラリを **.NET Compact Framework** アプリケーションから直接利用できるようにする、ラッパーライブラリです。

カメラライブラリを使用することにより、機種を意識することなく、アプリケーションのソースコード互換性を高めることができます。

カメラライブラリでは、機種を問わず、すべての関数を用意し、アプリケーションから見た「仮想マシン」としての振る舞いを提供します。

カメラライブラリの各関数は、アプリケーションからの要求に対して、対象のデバイス機能が制御できない場合は、「未サポートエラー」を返します。また搭載デバイスの機能差によって利用できないパラメータを設定した場合は、「パラメータエラー」を返します。

※ カメラライブラリは、アプリケーションのソースコード互換性の向上を目的としたライブラリであり、搭載デバイスの機能互換性を保障するものではありません。

「未サポートエラー」および「パラメータエラー」を正しく判定し、操作者に対して機能が未サポートである旨を通知する、あるいは処理そのものを無効としてください。

## 2. 動作環境

カメラライブラリの動作環境を以下に示します。

### 対象機種

DT-5200 / DT-5300 / IT-G500

### 対象 OS

- Microsoft Windows CE 5.0
- Microsoft Windows CE 6.0
- Microsoft Windows Embedded Compact 7
- Microsoft Windows Mobile 6.5
- Microsoft Windows Embedded Handheld 6.5

### 開発環境とプログラミング言語

表 2-1

開発環境	Visual C++	Visual Basic, Visual C#
Microsoft embedded Visual C++ Version 4.0 + SP4	○	-
Microsoft Visual Studio.NET 2003 + SP1	×	○
Microsoft Visual Studio 2005 + SP1	○	○
Microsoft Visual Studio 2008	○	○

(○:利用可、×:利用不可、-:機能なし)

### 提供ファイル

表 2-2

ファイル	Visual C++	Visual Basic, Visual C#
CameraLib.h	○	-
CameraLib.lib	○	-
CameraLib.dll	○	○
CameraLibNet.dll (クラスライブラリ)	-	○

(○:必要、-:不要)



## 使用方法

### Visual C++ の場合

- プログラムソース内に **CameraLib.h** をインクルードし、リンカの依存ファイルとして **CameraLib.lib** を指定してください
- **CameraLib.dll** は本体に内蔵されています。

### Visual Basic または Visual C# の場合

- **CameraLibNet.dll** をプロジェクトの参照に追加してください。
- **CameraLib.dll** は本体に内蔵されています。
- **CameraLibNet.dll** を実行モジュールと同じフォルダにコピーしてください。

## 名前空間とクラス

クラスライブラリ **CameraLibNet.dll** では、関数および定数の参照用として、下記のクラスが用意されています。

表 2-3

名前空間	クラス名	内容
CaLib	CameraLibNet.Api	関数参照用クラス
	CameraLibNet.Def	定数参照用クラス

クラス定義の詳細については、Microsoft Visual Studio で **CameraLibNet.dll** を参照設定し、オブジェクトブラウザで確認してください。

### 3. 構造体

カメラライブラリで提供する構造体を以下に示します。

表 3-1 構造体一覧

構造体名	内容	使用する関数
CAM_IMAGE_SET	JPEG 圧縮率など、取り込む静止画のオプション情報を格納する構造体	CAMSetImageSettings CAMGetImageSettings

#### 3.1 CAM\_IMAGE\_SET

JPEG 圧縮率など、取り込む静止画のオプション情報を格納します。

```
typedef struct _cam_image_settings
{
    int nQuality;
    int nJpegOpt;
} CAM_IMAGE_SET;
```

##### *nQuality*

JPEG ファイルの画質・圧縮率。

CAM_JPEG_LOW	: 画質=低、圧縮率=高
CAM_JPEG_NORMAL	: 画質=中、圧縮率=中(デフォルト)
CAM_JPEG_HIGH	: 画質=高、圧縮率=低
CAM_JPEG_HIGHEST	: 画質=最高、圧縮率=最低

##### *nJpegOpt*

JPEG ファイルのデータ保存オプション。

CAM_JPEG_BASELINE	: 上のラインから画像が表示されるファイル形式です。(デフォルト) DT-5300WM では必ずこの値を指定してください。
CAM_JPEG_PROGRESSIVE	: まず全体に粗い画像が表示され、徐々に詳細が表示されるファイル形式です。WEB ブラウザが対応していない場合があります。
CAM_JPEG_THUMB_QQVGA	: QQVGA(DT-5200:160x120、その他:120x160)サイズのサムネイル画像を埋め込みます。本画像部分はベースラインです。
CAM_JPEG_THUMB_QVGA	: QVGA(DT-5200:320x240、その他:240x320)サイズのサムネイル画像を埋め込みます。本画像部分はベースラインです。

	DT-5200	DT-5300	IT-G500
CAM_JPEG_LOW	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CAM_JPEG_NORMAL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CAM_JPEG_HIGH	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CAM_JPEG_HIGHEST	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CAM_JPEG_BASELINE	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CAM_JPEG_PROGRESSIVE	<input type="radio"/>	<input type="radio"/>	
CAM_JPEG_THUMB_QQVGA	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CAM_JPEG_THUMB_QVGA	<input type="radio"/>	<input type="radio"/>	

## 4. 関数

### 4.1 CAMOpen

カメラドライバをロードします。

```
[C++]  
int CAMOpen()
```

```
[Visual Basic]  
Public Shared Function CAMOpen() As Int32
```

```
[C#]  
public static Int32 CAMOpen()
```

#### 解説

カメラドライバをロードします。カメラドライバを使用する際には、必ずこの関数をコールしてください。

#### パラメータ

なし

#### 戻り値

CAM_SUCCESS	: 正常終了
CAM_OPENERROR	: カメラドライバが既にロードされているか、もしくはカメラを使用できません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSupport	: 未サポートエラー

## 4.2 CAMClose

カメラドライバをアンロードします。

```
[C++]  
int CAMClose()
```

```
[Visual Basic]  
Public Shared Function CAMClose() As Int32
```

```
[C#]  
public static Int32 CAMClose()
```

### 解説

カメラドライバをアンロードします。カメラドライバの使用を終了、またはアプリケーションプログラムを終了する際には、必ずこの関数をコールしてください。

### パラメータ

なし

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
FUNCTION_UNSupport	: 未サポートエラー

## 4.3 CAMStartPreview

プレビューを開始します。

```
[C++]
int CAMStartPreview(
    HWND hWnd,
    int nX,
    int nY,
    int nResSize
)
```

```
[Visual Basic]
Public Shared Function CAMStartPreview( _
    ByVal hWnd As IntPtr, _
    ByVal nX As Int32, _
    ByVal nY As Int32, _
    ByVal nResSize As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMStartPreview(
    IntPtr hWnd,
    Int32 nX,
    Int32 nY,
    Int32 nResSize
);
```

### 解説

プレビューを開始します。プレビューは `CAMStopPreview()`関数がコールされるまで行われます。  
Device Emulator では、`inf` ファイルで指定した静止画を表示します。詳細は、`CameraLib.inf`を参照してください。

### パラメータ

*hWnd*

プレビューを表示するウィンドウハンドルを指定します

*nX*

プレビューを表示する左上の X 座標を指定します。

*nY*

プレビューを表示する左上の Y 座標を指定します。

## *nResSize*

プレビューの画像解像度を指定します。

CAM\_VGA : VGA サイズ (480x640)  
CAM\_4PER9VGA : 4/9VGA サイズ (320x432)  
CAM\_QVGA : QVGA サイズ (240x320)  
CAM\_1PER9VGA : 1/9VGA サイズ (160x213)

OR 指定で画像の向き(回転方向)を指定することができます。

CAM\_ROTATE\_NONE : 回転なし  
CAM\_ROTATE\_90 : 時計回りに 90 度回転  
CAM\_ROTATE\_180 : 時計回りに 180 度回転  
CAM\_ROTATE\_270 : 時計回りに 270 度回転

OR 指定で横長プレビューを指定することができます。

横長プレビューを指定することにより、DT-5200 と同じように本体を縦向きにした状態で横長のプレビューを表示することができます。

ただし、カメラの画角は DT-5200 に比べて狭くなるので、撮影距離を調整してご使用ください。

CAM\_LANDSCAPE : 横長プレビュー表示

- 横長プレビューを指定する場合は、画像の向きに CAM\_ROTATE\_90、CAM\_ROTATE\_180、CAM\_ROTATE\_270 を指定することはできません。
- 横長プレビューを指定した場合、デジタルズーム機能の一部の倍率を指定することができません。詳細は CAMSetDigitalZoom を参照してください。

	DT-5200	DT-5300CE	DT-5300WM	IT-G500
CAM_VGA				○
CAM_4PER9VGA	○	○	○	○
CAM_QVGA	○	○	○	○
CAM_1PER9VGA	○	○	○	○
画像回転		○		○
横長プレビュー		○		○

## 戻り値

CAM\_SUCCESS : 正常終了  
CAM\_NOTOPENED : カメラドライバがロードされていません  
CAM\_OPENERERROR : プレビューが既に表示されています。  
CAM\_INVALID\_PARAM : 引数が正しくありません  
CAM\_SHOW\_ERROR : プレビューを表示できません  
CAM\_INTERNAL\_ERROR : ウィンドウハンドルや表示座標値を確認してください  
CAM\_INTERNAL\_ERROR : 内部エラーです  
Device Emulator では発生しません

CAM_NOT_ENOUGH_MEMORY	: プログラム実行用メモリが足りません プログラム実行用メモリの空きが <b>6MB</b> 以上あることを 確認してください <b>Device Emulator</b> では発生しません
CAM_INVALID_ACCESS	: イメージャが動作中です。 イメージャを終了してから実行してください。
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 注意

- アプリケーションを **Microsoft Foundation Class Library** のダイアログベースで開発される場合、本関数を **OnInitDialog** から呼び出さないでください。

```

SampleDlg::SampleDlg ()
{
    BEGIN_MESSAGE_MAP( SampleDlg, CDialog)
    ON_MESSAGE( WM_APP, OnDispPreview)
}

BOOL SampleDlg::OnInitDialog()
{
    CAMOpen();
    PostMessage( WM_APP);
}

LRESULT SampleDlg::OnDispPreview( WPARAM wParam, LPARAM lParam)
{
    CAMStartPreview( this->GetSafeHwnd(), 10, 10, CAM_QVGA | CAM_LANDSCAPE);
    return S_OK;
}

```

- プレビュー実行中に画面表示を回転させないでください。画面表示を回転する場合は、本関数を実行する前に回転させてください。



## 4.4 CAMStopPreview

プレビューを停止します。

```
[C++]  
int CAMStopPreview()
```

```
[Visual Basic]  
Public Shared Function CAMStopPreview() As Int32
```

```
[C#]  
public static Int32 CAMStopPreview()
```

### 解説

プレビューを停止します。

### パラメータ

なし

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_NOT_STARTED	: プレビューが開始されていません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSupport	: 未サポートエラー

## 4.5 CAMSingleCapture

静止画を取り込みます。

```
[C++]
int CAMSingleCapture(
    const TCHAR *lpzFileName,
    int nImageFormat,
    int nResSize
)
```

```
[Visual Basic]
Public Shared Function CAMSingleCapture( _
    ByVal lpzFileName As String, _
    ByVal nImageFormat As Int32, _
    ByVal nResSize As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSingleCapture(
    string lpzFileName,
    Int32 nImageFormat,
    Int32 nResSize
);
```

### 解説

1 画面分の静止画を取り込み、ファイルとして出力します。

カメラモジュールによる画像キャプチャが完了すると、シャッター音を鳴動し、WM\_USER + 0x560 をブロードキャストします。

Device Emulator では、inf ファイルで指定した画像を保存します。詳細は、CameraLib.inf を参照してください。

### パラメータ

#### *lpzFileName*

出力するファイル名をフルパスで指定します。

"¥¥My Documents"フォルダに"Image001.jpg"というファイル名で保存するには、TEXT("¥¥My Documents¥¥Image001.jpg")と指定します。

#### *nImageFormat*

画像ファイル形式を、以下の値で指定します

CAM_JPEG	: JPEG ファイル形式で保存します
CAM_BMP	: BMP ファイル形式で保存します

### *nResSize*

プレビューの画像解像度を指定します。

CAM_QUADSXGA	: Quad-SXGA サイズ	(1944x2592)
CAM_QUADXGA	: Quad-XGA サイズ	(1536x2048)
CAM_UXGA	: UXGA サイズ	(1200x1600)
CAM_QUADVGA	: QUAD-VGA サイズ	(960x1280)
CAM_1M	: 1M サイズ	(870x1160)
CAM_XGA	: XGA サイズ	(768x1024)
CAM_SVGA	: SVGA サイズ	(600x800)
CAM_VGA	: VGA サイズ	(480x640)
CAM_4PER9VGA	: 4/9VGA サイズ	(320x426)
CAM_QVGA	: 1/4VGA サイズ	(240x320)
CAM_1PER9VGA	: 1/9VGA サイズ	(213x160)

OR 指定で画像の向き(回転方向)を指定することができます。

CAM_ROTATE_NONE	: 回転なし
CAM_ROTATE_90	: 時計回りに 90 度回転
CAM_ROTATE_180	: 時計回りに 180 度回転
CAM_ROTATE_270	: 時計回りに 270 度回転

OR 指定で横長画像撮影を指定することができます。

横長画像撮影を指定することにより、DT-5200 と同じように本体を縦向きにした状態で横長の画像を撮影することができます。

ただし、カメラの画角は DT-5200 に比べて狭くなりますので、撮影距離を調整してご使用ください。

CAM\_LANDSCAPE : 横長画像撮影

- 横長画像撮影を指定する場合は、画像の向きに CAM\_ROTATE\_90、CAM\_ROTATE\_180、CAM\_ROTATE\_270 を指定することはできません。
- 横長画像撮影を指定した場合に撮影できる画像サイズは、QVGA、VGA、SVGA、XGA です。Quad VGA、UXGA を撮影することはできませんのでご注意ください。
- 横長画像撮影を指定した場合、デジタルズーム機能の一部の倍率を指定することができません。詳細は CAMSetDigitalZoom を参照してください。

	DT-5200	DT-5300CE	DT-5300WM	IT-G500
CAM_JPEG	○	○	○	○
CAM_BMP	○	○	○	
CAM_QUADSXGA				○
CAM_QUADXGA				○
CAM_UXGA		○	○	○
CAM_QUADVGA		○	○	○
CAM_1M	○			
CAM_XGA	○	○	○	○
CAM_SVGA	○	○	○	○
CAM_VGA	○	○	○	○
CAM_4PER9VGA	○			
CAM_QVGA		○	○	○
CAM_1PER9VGA	○			
画像回転		○		○
横長画像撮影		○		○

## 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_NOT_STARTED	: プレビューが開始されていません
CAM_FILE_ERROR	: 画像ファイルを保存できません パス名やディスク残り容量、ストレージデバイスの有無などを 確認してください
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
CAM_NOT_ENOUGH_MEMORY	: プログラム実行用メモリが足りません Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 注意

- DT-5300 では、フラッシュ LED 付近の温度が高温になった場合、CAM\_INTERNAL\_ERROR を返し、撮影に失敗します。
- 大きな画像(CAM\_QUADSXGA、CAM\_QUADXGA など)を撮影する場合は、メモリの空き容量にご注意ください。プレビュー開始後、撮影直前(CAMSingleCapture 関数のコール直前)に、アプリケーションプロセス内に以下の使用可能領域(仮想メモリ)があることを確認してください。
 

CAM_QUADSXGA	: 24MB 以上
CAM_QUADXGA	: 20MB 以上
CAM_UXGA	: 17MB 以上
その他の解像度	: 14MB 以上

## 4.6 CAMSetImageSettings

静止画オプションを設定します。

```
[C++]
int CAMSetImageSettings(
    CAM_IMAGE_SET *lpImageSettings
)
```

```
[Visual Basic]
Public Shared Function CAMSetImageSettings( _
    ByRef lpImageSettings As CameraLibNet.CAM_IMAGE_SET& _
) As Int32
```

```
[C#]
public static Int32 CAMSetImageSettings(
    ref CameraLibNet.CAM_IMAGE_SET& lpImageSettings
);
```

### 解説

JPEG 圧縮率など、取り込む静止画のオプションを指定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetImageSettings 関数を実行することにより、設定値を確認することができます。

### パラメータ

*lpImageSettings*  
CAM\_IMAGE\_SET を指定します。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです
	Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.7 CAMGetImageSettings

静止画オプションを取得します。

```
[C++]
int CAMGetImageSettings(
    CAM_IMAGE_SET *lpImageSettings
)
```

```
[Visual Basic]
Public Shared Function CAMGetImageSettings( _
    ByRef lpImageSettings As CameraLibNet.CAM_IMAGE_SET& _
) As Int32
```

```
[C#]
public static Int32 CAMGetImageSettings(
    ref CameraLibNet.CAM_IMAGE_SET& lpImageSettings
);
```

### 解説

JPEG 圧縮率など、取り込む静止画のオプションを取得します。

### パラメータ

*lpImageSettings*

CAM\_IMAGE\_SET を指定します。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数に NULL が指定されました
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.8 CAMContinuousCapture

連写画像を取り込みます。

```
[C++]
int CAMContinuousCapture(
    Const TCHAR *lpszFileName,
    int nImageFormat,
    int nNumOfFiles,
    int nFrameRate
)
```

```
[Visual Basic]
Public Shared Function CAMContinuousCapture( _
    ByVal lpszFileName As String, _
    ByVal nImageFormat As Int32, _
    ByVal nNumOfFiles As Int32, _
    ByVal nFrameRate As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMContinuousCapture(
    string lpszFileName,
    Int32 nImageFormat,
    Int32 nNumOfFiles,
    Int32 nFrameRate
);
```

### 解説

連写画像を取り込み、指定した枚数分のファイルとして出力します。

### パラメータ

#### *lpszFileName*

出力するファイルのベース名をフルパスで指定します。

本関数は、指定されたファイル名に 2 桁の番号を付加して出力します。TEXT("¥¥My Documents¥¥Image.jpg")と指定された場合は、"My Documents"ディレクトリに、"Image01.jpg"、"Image02.jpg"...ファイルを作成します。

#### *nImageFormat*

画像ファイル形式を、以下の値で指定します

CAM_JPEG	: JPEG ファイル形式で保存します
CAM_BMP	: BMP ファイル形式で保存します

#### *nNumOfFiles*

取り込む連写画像の枚数を 1～20 の範囲で指定します

#### *nFrameRate*

取り込む連写画像のフレームレートを、以下の値で指定します

CAM\_5FPS : 5 フレーム/秒  
CAM\_15FPS : 15 フレーム/秒

#### 戻り値

CAM\_SUCCESS : 正常終了  
CAM\_NOTOPENED : カメラドライバがロードされていません  
CAM\_INVALID\_PARAM : 引数が正しくありません  
CAM\_NOT\_STARTED : プレビューが開始されていません  
CAM\_FILE\_ERROR : 画像ファイルを保存できません  
パス名やディスク残り容量、ストレージデバイスの有無などを確認してください  
CAM\_INTERNAL\_ERROR : 内部エラーです  
Device Emulator では発生しません  
FUNCTION\_UNSUPPORTED : 未サポートエラー



## 4.9 CAMSetLightMode

撮影モードを設定します。

```
[C++]
int CAMSetLightMode(
    int nLightMode
)
```

```
[Visual Basic]
Public Shared Function CAMSetLightMode( _
    ByVal nLightMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetLightMode(
    Int32 nLightMode
);
```

### 解説

撮影モード(ホワイトバランス)を設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetLightMode 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *nLightMode*

撮影モードを、以下の値で設定します。

- CAM\_LIGHT\_AUTO : 自動(デフォルト)
  - CAM\_OUTDOOR : 屋外(自然光)
  - CAM\_FLUORESCENT : 蛍光灯
  - CAM\_INCANDESCENCE : 白熱灯
  - CAM\_DIMLIGHT※ : 暗所
- ※ DT-5200 では指定できません

### 戻り値

- CAM\_SUCCESS : 正常終了
- CAM\_NOTOPENED : カメラドライバがロードされていません
- CAM\_INVALID\_PARAM : 引数が正しくありません
- CAM\_INTERNAL\_ERROR : 内部エラーです  
Device Emulator では発生しません
- FUNCTION\_UNSUPPORTED : 未サポートエラー

## 4.10 CAMGetLightMode

撮影モードを取得します。

```
[C++]  
int CAMGetLightMode()
```

```
[Visual Basic]  
Public Shared Function CAMGetLightMode() As Int32
```

```
[C#]  
public static Int32 CAMGetLightMode()
```

### 解説

現在の撮影モード(ホワイトバランス)を取得します。

### パラメータ

なし

### 戻り値

現在の撮影モードを取得します。取得する値は **CAMSetLightMode** 関数を参照してください。  
または、

<b>CAM_NOTOPENED</b>	: カメラドライバをロードしていません
<b>FUNCTION_UNSupport</b>	: 未サポートエラー

## 4.11 CAMSetDigitalZoom

デジタルズームを設定します。

```
[C++]
int CAMSetDigitalZoom(
    int nZoom
)
```

```
[Visual Basic]
Public Shared Function CAMSetDigitalZoom( _
    ByVal nZoom As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetDigitalZoom(
    Int32 nZoom
);
```

### 解説

デジタルズームを設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetDigitalZoom 関数を実行することにより、設定値を確認することができます。

### パラメータ

*nZoom*

デジタルズームの倍率を、以下の値で指定します

CAM_ZOOM_NONE	: 等倍(デジタルズームを使用しない)。(デフォルト)
CAM_ZOOM_15	: 1.5 倍
CAM_ZOOM_20	: 2.0 倍
CAM_ZOOM_30	: 3.0 倍

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 補足

下記の×の設定で撮影を行うと、デジタルズームなし(1.0 倍)で撮影を行います。

### DT-5200

倍率	1/9VGA	1/4VGA	4/9VGA	VGA	SVGA	XGA
1.5 倍	○	○	○	○	×	×
2.0 倍	○	○	○	×	×	×
3.0 倍	○	○	○	×	×	×

### DT-5300

倍率	1/4VGA	VGA	SVGA	XGA	Quad-VGA	UXGA
1.5 倍	○	○	○	○	×	×
2.0 倍	○	○	○	×	×	×
3.0 倍	○	×	×	×	×	×

横長プレビューを使用する場合(プレビューの解像度が 1/9VGA、1/4VGA の場合)

倍率	1/4VGA	VGA	SVGA	XGA	Quad VGA	UXGA
1.5 倍	○	○	○	×	×	×
2.0 倍	×	×	×	×	×	×
3.0 倍	×	×	×	×	×	×

1/9VGA もしくは QVGA の横長プレビュー表示中に 2.0 倍、3.0 倍を指定すると、CAM\_INVALID\_PARAM が返ります。

横長プレビューを使用する場合(プレビューの解像度が 4/9VGA の場合)

倍率	1/4VGA	VGA	SVGA	XGA	Quad VGA	UXGA
1.5 倍	×	×	×	×	×	×
2.0 倍	×	×	×	×	×	×
3.0 倍	×	×	×	×	×	×

4/9VGA の横長プレビュー表示中に 1.5 倍、2.0 倍、3.0 倍を指定すると、CAM\_INVALID\_PARAM が返ります。

### DT-5300WM

倍率	1/4VGA	VGA	SVGA	XGA	Quad-VGA	UXGA
1.5 倍	○	○	○	○	×	×
2.0 倍	○	○	○	×	×	×

### IT-G500

倍率	1/4VGA	VGA	SVGA	XGA	Quad VGA	UXGA	Quad XGA	Quad SXGA
1.5 倍	○	○	○	○	○	○	×	×
2.0 倍	○	○	○	○	○	×	×	×
3.0 倍	○	○	○	×	×	×	×	×

横長プレビューを使用する場合

倍率	1/4VGA	VGA	SVGA	XGA	Quad VGA	UXGA	Quad XGA	Quad SXGA
1.5 倍	○	○	○	○	○	×	×	×
2.0 倍	○	○	○	×	×	×	×	×
3.0 倍	○	○	×	×	×	×	×	×

## 4.12 CAMGetDigitalZoom

デジタルズームを取得します。

```
[C++]  
int CAMGetDigitalZoom()
```

```
[Visual Basic]  
Public Shared Function CAMGetDigitalZoom() As Int32
```

```
[C#]  
public static Int32 CAMGetDigitalZoom()
```

### 解説

現在のデジタルズーム設定を取得します。

### パラメータ

なし

### 戻り値

現在のデジタルズーム設定を取得します。取得する値は CAMSetDigitalZoom 関数を参照してください。

CAM_NOTOPENED	: カメラドライバをロードしていません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.13 CAMSetSensitivity

明るさを設定します。

```
[C++]
int CAMSetSensitivity(
    int nSensitivity
)
```

```
[Visual Basic]
Public Shared Function CAMSetSensitivity( _
    ByVal nSensitivity As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetSensitivity(
    Int32 nSensitivity
);
```

### 解説

明るさ(露出補正)を設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetSensitivity 関数を実行することにより、設定値を確認することができます。

### パラメータ

*nSensitivity*

明るさの数値を 0(暗い)～24(明るい)の範囲で指定します。デフォルトは 12 です。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです
	Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.14 CAMGetSensitivity

明るさを取得します。

```
[C++]  
int CAMGetSensitivity()
```

```
[Visual Basic]  
Public Shared Function CAMGetSensitivity() As Int32
```

```
[C#]  
public static Int32 CAMGetSensitivity()
```

### 解説

現在の明るさ(露出補正值)を取得します。

### パラメータ

なし

### 戻り値

現在の明るさ(露出補正值)を取得します。取得する値は **CAMSetSensitivity** 関数を参照してください。  
または、以下の値が返ります。

<b>CAM_NOTOPENED</b>	: カメラドライバがロードされていません
<b>FUNCTION_UNSUPPORTED</b>	: 未サポートエラー



## 4.15 CAMRedrawPreview

プレビュー画面の再描画を行います。

```
[C++]  
int CAMRedrawPreview()
```

```
[Visual Basic]  
Public Shared Function CAMRedrawPreview() As Int32
```

```
[C#]  
public static Int32 CAMRedrawPreview()
```

### 解説

プレビュー画面の再描画を行います。

この関数は、プレビュー画面上にメッセージボックス等が表示され、再描画が必要なときにコールします。通常はアプリケーションプログラムに WM\_PAINT メッセージが送られてきたときにコールしてください。

### パラメータ

なし

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.16 CAMSingleCaptureEx

1 画面分の静止画を取り込み、静止画データをメモリに出力します。

```
[C++]
int CAMSingleCaptureEx(
    LPBYTE pbImage,
    int nResSize,
    int nColorSpace,
    int nReserved
)
```

```
[Visual Basic]
Public Shared Function CAMSingleCaptureEx( _
    ByRef pbImage () As Byte _
    ByVal nResSize As Int32 _
    ByVal nColorSpace As Int32 _
    ByVal nReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSingleCaptureEx(
    ref byte pbImage,
    Int32 nResSize,
    Int32 nColorSpace,
    Int32 nReserved
)
```

### 解説

1 画面分の静止画を取り込み、静止画データをメモリに出力します。

本関数は、CAMStartPreview 関数をコールした後に使用してください。

本関数をコールすると、プレビューはストップします。プレビューを再開する場合は、CAMStartPreview 関数をコールしてください。

Device Emulator では、inf ファイルで指定した画像を表示します。詳細は、CameraLib.inf を参照してください。

### パラメータ

#### *lbImage*

画像データを格納する領域の先頭アドレスのポインタを指定します。

受け取る画像サイズに応じて、次のエリアを確保してください。

	YUV422	RGB888
CAM_QUADSXGA	: 10,077,696	15,116,544
CAM_QUADXGA	: 6,291,456	9,437,184
CAM_UXGA	: 3,840,000	5,760,000
CAM_QUADVGA	: 2,457,600	3,686,400
CAM_1M	: 2,018,400	3,027,600
CAM_XGA	: 1,572,864	2,359,296
CAM_SVGA	: 960,000	1,440,000

CAM_VGA	: 614,400	921,600
CAM_4PER9VGA	: 276,480	424,720
CAM_QVGA	: 153,600	230,400
CAM_1PER9VGA	: 69,120	103,680

#### *nResSize*

プレビューの画像解像度を指定します。

CAM_QUADSXGA	: Quad-SXGA サイズ	(1944x2592)
CAM_QUADXGA	: Quad-XGA サイズ	(1536x2048)
CAM_UXGA	: UXGA サイズ	(1200x1600)
CAM_QUADVGA	: QUAD-VGA サイズ	(960x1280)
CAM_1M	: 1M サイズ	(870x1160)
CAM_XGA	: XGA サイズ	(768x1024)
CAM_SVGA	: SVGA サイズ	(600x800)
CAM_VGA	: VGA サイズ	(480x640)
CAM_4PER9VGA	: 4/9VGA サイズ	(320x426)
CAM_QVGA	: 1/4VGA サイズ	(240x320)
CAM_1PER9VGA	: 1/9VGA サイズ	(213x160)

OR 指定で画像の向き(回転方向)を指定することができます。

CAM_ROTATE_NONE	: 回転なし
CAM_ROTATE_90	: 時計回りに 90 度回転
CAM_ROTATE_180	: 時計回りに 180 度回転
CAM_ROTATE_270	: 時計回りに 270 度回転

OR 指定で横長画像撮影を指定することができます。

横長画像撮影を指定することにより、DT-5200 と同じように本体を縦向きにした状態で横長の画像を撮影することができます。

ただし、カメラの画角は DT-5200 に比べて狭くなりますので、撮影距離を調整してご使用ください。

CAM_LANDSCAPE	: 横長画像撮影
---------------	----------

- 横長画像撮影を指定する場合は、画像の向きに CAM\_ROTATE\_90、CAM\_ROTATE\_180、CAM\_ROTATE\_270 を指定することはできません。
- 横長画像撮影を指定した場合に撮影できる画像サイズは、QVGA、VGA、SVGA、XGA です。Quad VGA、UXGA を撮影することはできませんのでご注意ください。
- 横長画像撮影を指定した場合、デジタルズーム機能の一部の倍率を指定することができません。詳細は CAMSetDigitalZoom を参照してください。

	DT-5200	DT-5300CE	DT-5300WM	IT-G500
CAM_YUV422	○	○	○	○
CAM_RGB888	○	○	○	○
CAM_QUADSXGA				○
CAM_QUADXGA				○
CAM_UXGA		○	○	○
CAM_QUADVGA		○	○	○

CAM_1M	○			
CAM_XGA	○	○	○	○
CAM_SVGA	○	○	○	○
CAM_VGA	○	○	○	○
CAM_4PER9VGA	○			
CAM_QVGA		○	○	○
CAM_1PER9VGA	○			
画像回転		○		○
横長画像撮影		○		○

### *nColorSpace*

出力される画像のカラースペースを指定します。

CAM\_YUV422 : YUV422 カラースペース  
CAM\_RGB888 : RGB888(24bit)カラースペース

### *nReserved*

現在のバージョンではこの引数は無視されます。

### 戻り値

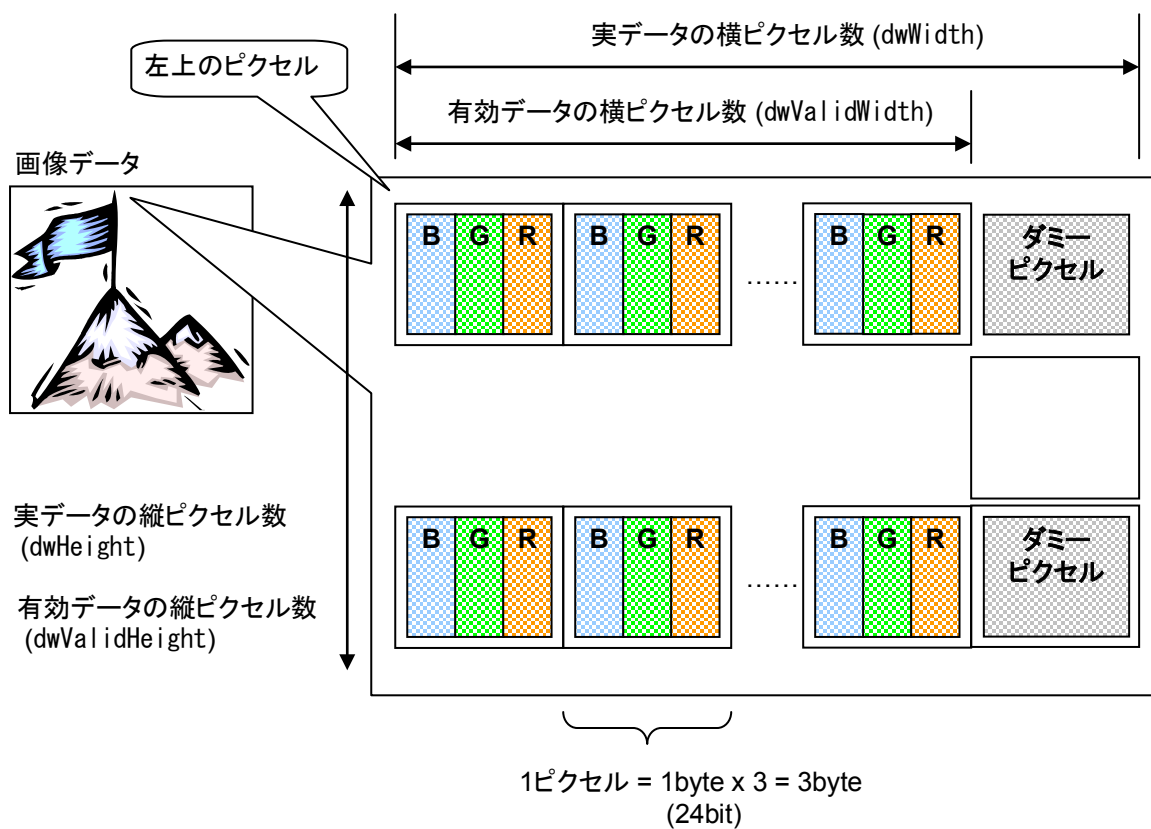
CAM\_SUCCESS : 正常終了  
CAM\_NOTOPENED : カメラドライバがロードされていません  
CAM\_INVALID\_PARAM : 引数が正しくありません  
CAM\_NOT\_STARTED : プレビューが開始されていません  
CAM\_FILE\_ERROR : 画像ファイルを保存できません  
パス名やディスク残り容量、ストレージデバイスの有無などを確認してください  
CAM\_INTERNAL\_ERROR : 内部エラーです  
Device Emulator では発生しません  
CAM\_NOT\_ENOUGH\_MEMORY : プログラム実行用メモリが足りません  
Device Emulator では発生しません  
FUNCTION\_UNSUPPORT : 未サポートエラー

### 注意

- DT-5300 では、フラッシュ LED 付近の温度が高温になった場合、CAM\_INTERNAL\_ERROR を返し、撮影に失敗します。
- 大きな画像(CAM\_QUADSXGA、CAM\_QUADXGA など)を撮影する場合は、メモリの空き容量にご注意ください。プレビュー開始後、撮影直前(CAMSingleCaptureEx 関数のコール直前)に、アプリケーションプロセス内に以下の使用可能領域(仮想メモリ)があることを確認してください。  
CAM\_QUADSXGA : 24MB 以上  
CAM\_QUADXGA : 20MB 以上  
CAM\_UXGA : 17MB 以上  
その他の解像度 : 14MB 以上

## ■RGB888 データ

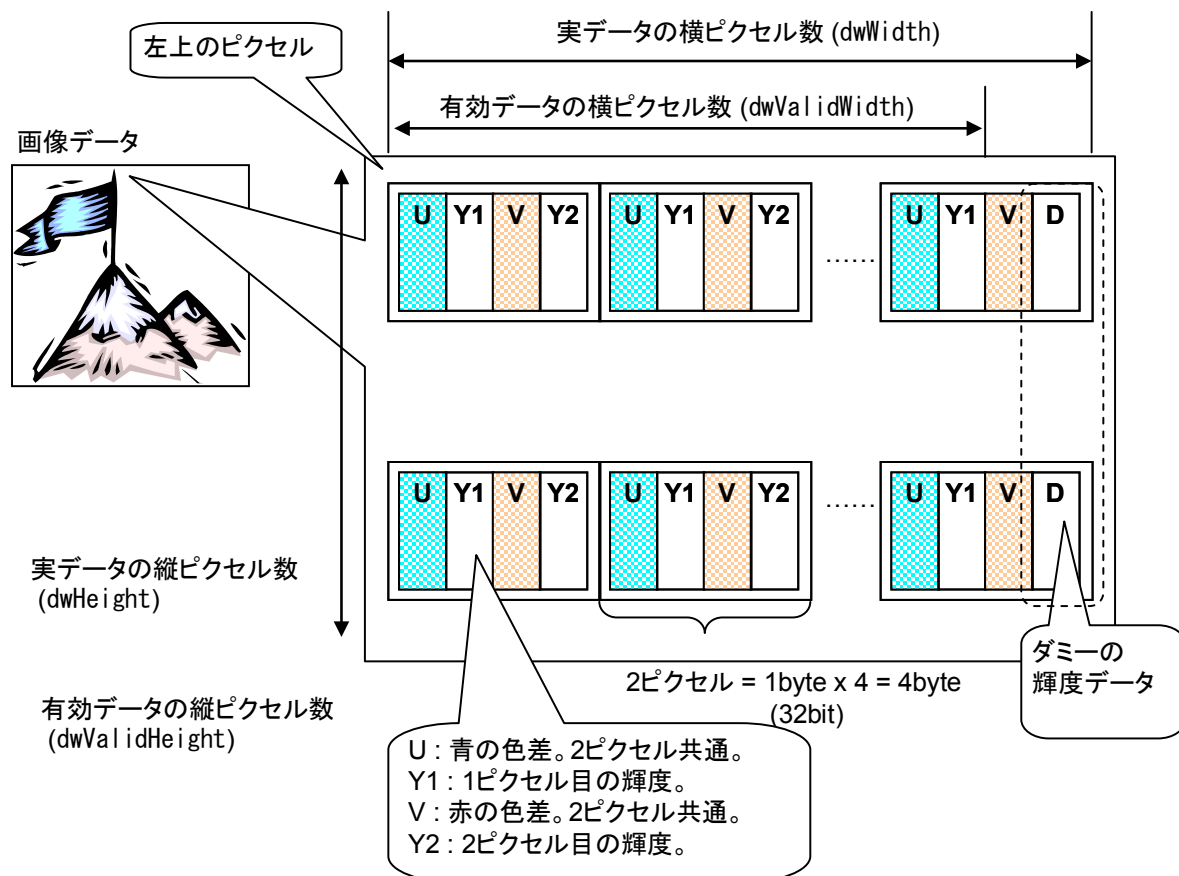
1 ピクセルを B、G、R 各 1byte (8bit) で表したデータフォーマットです。データには縦・横ともにダミーピクセルを含むことができます。このため、ビットマップファイルのようにダミーデータを含んだデータもエンコードすることができます(※)。



(※) 24bit ビットマップファイルは RGB888 データで構成されますが、横幅の実サイズが 4 の倍数でなければなりません。このため、画像の横幅が 4 の倍数でない場合は、上記のようにダミーデータを挿入し、横幅のサイズを調整する必要があります。

## ■YUV422 データ

2 ピクセルを、輝度(Y)1byte×2、青の色差(U)1byte、赤の色差(V)1byte で表したデータフォーマットです。先頭のデータが画像の左上になります。



データには縦・横ともにダミーピクセルを含むことができます。YUV422 フォーマットは横幅が偶数でなければなりませんが、これにより横幅が奇数の画像データもエンコードすることができます。

## 4.17 CAMIlluminationOn

照明 LED の点灯／消灯を行います。

```
[C++]
int CAMIlluminationOn(
    BOOL bEnable
)
```

```
[Visual Basic]
Public Shared Function CAMIlluminationOn ( _
    ByVal bEnable As Boolean _
) As Int32
```

```
[C#]
public static Int32 CAMIlluminationOn (
    BOOL bEnable
);
```

### 解説

点灯時は CAMSetIllumination 関数で設定した明るさで点灯します。  
Device Emulator では、引数チェックのみを行い、何も動作しません。

### パラメータ

#### *bEnable*

照明 LED の点灯／消灯を指定します。

TRUE : 照明 LED を点灯  
FALSE : 照明 LED を消灯

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_ACCESS	: 排他エラーです Device Emulator では発生しません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 注意

イメージャデバイスをオープン中に本関数を実行すると、排他エラーが返ります。イメージャデバイスをクローズした後に、再度本関数を実行してください。

DT-5300 では、フラッシュ LED 付近の温度が高温になった場合、CAM\_INTERNAL\_ERROR を返し、LED の点灯に失敗します。

## 4.18 CAMSetIllumination

照明 LED およびフラッシュの明るさを指定します。

```
[C++]
int CAMSetIllumination(
    DWORD dwBrightness
)
```

```
[Visual Basic]
Public Shared Function CAMSetIllumination ( _
    ByVal dwBrightness As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetIllumination (
    Int32 dwBrightness
);
```

### 解説

照明 LED およびフラッシュの明るさを指定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetIllumination 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *dwBrightness*

0(暗い)～100(明るい)の範囲で、照明 LED の明るさを指定します。

デフォルトは DT-5200 では 100、DT-5300/IT-G500 では 80 です。

IT-G500 では、明るさを切り替えることはできません。0 を指定するとフラッシュは点灯しません。1～100 を指定するとフラッシュが点灯します。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー



## 4.19 CAMGetIllumination

現在の照明 LED およびフラッシュの明るさを取得します。

```
[C++]
int CAMGetIllumination(
    LPDWORD pBrightness
)
```

```
[Visual Basic]
Public Shared Function CAMGetIllumination ( _
    ByRef pBrightness As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMGetIllumination (
    ref Int32 pBrightness
);
```

### 解説

現在の照明 LED およびフラッシュの明るさを取得します。

### パラメータ

*pBrightness*

取得する値は CAMSetIllumination 関数を参照してください。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.20 CAMSetIris

シャッターの絞りを設定します。

```
[C++]
int CAMSetIris(
    DOWRD dwIris
)
```

```
[Visual Basic]
Public Shared Function CAMSetIris ( _
    ByVal dwIris As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetIris (
    Int32 dwIris
);
```

### 解説

F=3.5 と F=7.0 の 2 段階を設定することができます。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetIris 関数を実行することにより、設定値を確認することができます。

### パラメータ

*dwIris*

絞りを設定します。

CAM_IRIS_35	: F=3.5 (デフォルト)
CAM_IRIS_70	: F=7.0

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.21 CAMGetIris

現在のシャッター絞り設定を取得します。

```
[C++]
int CAMGetIris(
    LPDWORD pIris
)
```

```
[Visual Basic]
Public Shared Function CAMGetIris( _
    ByRef pIris As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMGetIris (
    ref Int32 pIris
);
```

### 解説

現在のシャッター絞り設定を取得します。

### パラメータ

*pIris*

取得する値は CAMSetIris 関数を参照してください。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです
	Device Emulator では発生しません
FUNCTION_UNSupport	: 未サポートエラー

# 4.22 CAMSetFlickerless

フリッカレスモードを設定します。

```
[C++]
int CAMSetFlickerless(
    DWORD dwFlickerMode
)
```

```
[Visual Basic]
Public Shared Function CAMSetFlickerless ( _
    ByVal dwFlickerMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetFlickerless (
    Int32 dwFlickerMode
);
```

解説

蛍光灯(インバーター照明を除く、50Hz/60Hz 交流電源の蛍光灯)下でプレビューを表示すると、フリッカの影響を受けて画面がちらつくことがあります。本関数でフリッカレスモードを指定すると、ちらつきを軽減することができます。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetFlickerless 関数を実行することにより、設定値を確認することができます。

パラメータ

dwFlickerMode

フリッカレスモードを指定します。

- CAM\_FLICKER\_NONE : フリッカなし
- CAM\_FLICKER\_AUTO : フリッカレス自動
- CAM\_FLICKER\_50 : 50Hz フリッカレスモード
- CAM\_FLICKER\_60 : 60Hz フリッカレスモード

	DT-5200	DT-5300CE	DT-5300WM	IT-G500
CAM_FLICKER_NONE		○	○	
CAM_FLICKER_AUTO		○	○	○
CAM_FLICKER_50	○	○	○	
CAM_FLICKER_60	○	○	○	

## 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UN SUPPORT	: 未サポートエラー

## 4.23 CAMGetFlickerless

現在のフリッカレスモードを取得します。

```
[C++]
int CAMGetFlickerless(
    LPDWORD lpFlickerMode
)
```

```
[Visual Basic]
Public Shared Function CAMGetFlickerless ( _
    ByRef lpFlickerMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMGetFlickerless (
    ref Int32 lpFlickerMode
);
```

### 解説

現在のフリッカレスモードを取得します。

### パラメータ

*lpFlickerMode*

取得する値は CAMSetFlickerless 関数を参照してください。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです
	Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.24 CAMSetDigitalFlashMode

デジタルフラッシュモードを設定します。

```
[C++]
int CAMSetDigitalFlashMode (
    BOOL bDFMode,
    int nNumber,
    int nCaptureType,
    int nReserved
)
```

```
[Visual Basic]
Public Shared Function CAMSetDigitalFlashMode ( _
    ByVal bDFMode As Boolean _
    ByVal nNumber As Int32 _
    ByVal nCaptureType As Int32 _
    ByVal nReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetDigitalFlashMode (
    bool bDFMode,
    Int32 nNumber,
    Int32 nCaptureType,
    Int32 nReserved
);
```

### 解説

デジタルフラッシュモードを設定すると、CAMSingleCapture 関数および CAMSingleCaptureEx 関数コール時にデジタルフラッシュ撮影が行われます。デジタルフラッシュ撮影では、*nNumber* で指定した枚数の連写撮影が行われ、1 枚の画像に合成されて出力されます。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetDigitalFlashMode 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *bDFMode*

デジタルフラッシュモードを設定します。

FALSE : 通常モード(デフォルト)  
TRUE : デジタルフラッシュモード

#### *nNumber*

1～8 枚の範囲で、デジタルフラッシュ連写枚数を指定します。デフォルトは 4 枚です。*bDFMode* に TRUE を設定して CAMSingleCaptureEx 関数をコールすると、*nNumber* に指定した枚数の連写撮影が行われます。*bDFMode* に FALSE を設定すると本引数は無視されます。

#### *nCaptureType*

デジタルフラッシュ連写のキャプチャタイプを設定します。CAM\_CAP\_ESHUTTER を指定すると、

CAMSingleCaptureEx 関数で高速連写が行われますが、出力される画像サイズは QVGA のみとなります。CAM\_CAP\_MSHUTTER を指定すると、連写スピードは低速ですが、QVGA のほかに VGA の画像出力が可能です。bDFMode に FALSE を設定すると本引数は無視されます。

CAM_CAP_ESHUTTER	: 電子シャッター(高速、QVGA のみ)でのキャプチャ
CAM_CAP_MSHUTTER	: メカシャッター(低速、QVGA/VGA)でのキャプチャ(デフォルト)

*nReserved*

現在のバージョンではこの引数は使用されません。

**戻り値**

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー



## 4.25 CAMGetDigitalFlashMode

デジタルフラッシュモードを取得します。

```
[C++]
int CAMGetDigitalFlashMode (
    LPBOOL  /bDFMode,
    int*    /pNumber,
    int*    /pCaptureType,
    int*    /pReserved
)
```

```
[Visual Basic]
Public Shared Function CAMGetDigitalFlashMode ( _
    ByRef /bDFMode As Boolean _
    ByRef /pNumber As Int32 _
    ByRef /pCaptureType As Int32 _
    ByRef /pReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetDigitalFlashMode (
    ref bool /bDFMode,
    ref Int32 /pNumber,
    ref Int32 /pCaptureType,
    ref Int32 /pReserved
);
```

### 解説

現在のデジタルフラッシュモードを取得します。

### パラメータ

*/bDFMode*

取得する値は CAMSetDigitalFlashMode 関数を参照してください。

*/pNumber*

取得する値は CAMSetDigitalFlashMode 関数を参照してください。

*/pCaptureType*

取得する値は CAMSetDigitalFlashMode 関数を参照してください。

*/pReserved*

現在のバージョンではこの引数は使用されません。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません

---

CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSupport	: 未サポートエラー

## 4.26 CAMSetFlash

フラッシュの使用／不使用の切り替えを行います。

```
[C++]
int CAMSetFlash (
    DWORD    dwFlash,
    DWORD    dwReserved
)
```

```
[Visual Basic]
Public Shared Function CAMSetFlash ( _
    ByVal dwFlash As Int32, _
    ByVal dwReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetFlash (
    Int32 dwFlash,
    Int32 dwReserved
);
```

### 解説

フラッシュの使用／不使用の切り替えを行います。**CAM\_FLASH\_ENABLE** を指定すると、静止画撮影時にフラッシュが点灯します。点灯時は **CAMSetIllumination** 関数で設定した明るさで点灯します。**Device Emulator** では、設定値を内部変数として格納するため、何も動作しませんが、**CAMSetFlash** 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *dwFlash*

フラッシュの使用／不使用を指定します。

- |                          |  |
|--------------------------|--|
| <b>CAM_FLASH_ENABLE</b>  | : 静止画撮影時にフラッシュが点灯します                       |
| <b>CAM_FLASH_DISABLE</b> | : 静止画撮影時にフラッシュは点灯しません。(デフォルト)              |
| <b>CAM_FLASH_AUTO</b>    | : 暗い場所での撮影時にフラッシュを点灯します ( <b>IT-G500</b> ) |

#### *dwReserved*

現在のバージョンではこの引数は使用されません。**NULL** を指定してください。

### 戻り値

- |                             |  |
|-----------------------------|--|
| <b>CAM_SUCCESS</b>          | : 正常終了                                       |
| <b>CAM_NOTOPENED</b>        | : カメラドライバがロードされていません                         |
| <b>CAM_INVALID_PARAM</b>    | : 引数が正しくありません                                |
| <b>CAM_INTERNAL_ERROR</b>   | : 内部エラーです<br><b>Device Emulator</b> では発生しません |
| <b>FUNCTION_UNSUPPORTED</b> | : 未サポートエラー                                   |

## 4.27 CAMGetFlash

現在のフラッシュモードを取得します。

```
[C++]
int CAMGetFlash (
    LPDWORD pFlash,
    LPDWORD pReserved
)
```

```
[Visual Basic]
Public Shared Function CAMGetFlash ( _
    ByRef pFlash As Int32, _
    ByRef pReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMGetFlash (
    ref Int32 pFlash,
    ref Int32 pReserved
);
```

### 解説

現在のフラッシュモードを取得します。

### パラメータ

#### *pFlash*

フラッシュモードを取得します。

CAM_FLASH_ENABLE	: 静止画撮影時にフラッシュが点灯します
CAM_FLASH_DISABLE	: 静止画撮影時にフラッシュは点灯しません。(デフォルト)

#### *pReserved*

現在のバージョンではこの引数は使用されません。NULL を指定してください。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.28 CAMSetFocus

焦点モードおよび焦点位置を設定します。

```
[C++]
int CAMSetFocus (
    DWORD    dwFocus,
    DWORD    dwFocusMode
)
```

```
[Visual Basic]
Public Shared Function CAMSetFocus ( _
    ByVal dwFocus As Int32, _
    ByVal dwFocusMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetFocus (
    Int32 dwFocus,
    Int32 dwFocusMode
);
```

### 解説

焦点モードおよび焦点位置を設定します。

プレビュー実行中に本関数をコールすると、ただちに指定した焦点位置にピントを合わせます。  
プレビュー実行前にコールした場合は、プレビュー開始時に、指定した焦点位置が反映されます。

連続オートフォーカスモードでは、プレビュー表示中は常にオートフォーカスが動作し、ピントが合った状態になります。  
プレビュー実行前、実行中どちらでも本関数を実行することができます。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetFocus 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *dwFocus*

固定焦点モード (CAM\_FOCUS\_NORMAL) の焦点位置を 1 (近接) ～10 (遠距離) の範囲で指定します。

連続オートフォーカスモード (CAM\_FOCUS\_CAF) を使用する場合は 0 を指定してください。

デフォルト値は、10 です。

#### *dwFocusMode*

焦点モードを設定します。(IT-G500 のみ有効です。)

CAM_FOCUS_NORMAL	: 固定焦点モード (デフォルト)
CAM_FOCUS_CAF	: 連続オートフォーカスモード

## 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UN SUPPORT	: 未サポートエラー

## 4.29 CAMGetFocus

CAMSetFocus 関数で設定された焦点位置を取得します。

```
[C++]
int CAMGetFocus (
    LPDWORD pFocus,
    LPDWORD pFocusMode
)
```

```
[Visual Basic]
Public Shared Function CAMGetFocus ( _
    ByRef pFocus As Int32, _
    ByRef pFocusMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMGetFocus (
    ref Int32 pFocus,
    ref Int32 pFocusMode
);
```

### 解説

CAMSetFocus 関数で設定された焦点位置を取得します。

### パラメータ

#### *pFocus*

CAMSetFocus 関数で設定された焦点位置を、1(近接)～10(遠距離)の範囲で取得します。

#### *pFocusMode*

CAMSetFocus 関数で設定された焦点モードを取得します。

CAM_FOCUS_NORMAL	: 固定焦点モード (デフォルト)
CAM_FOCUS_CAF	: 連続オートフォーカスモード

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.30 CAMGetFocusStatus

現在の焦点位置を取得します。

```
[C++]
int CAMGetFocusStatus (
    LPDWORD pFocusStatus
)
```

```
[Visual Basic]
Public Shared Function CAMGetFocusStatus ( _
    ByRef pFocusStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMGetFocusStatus (
    ref Int32 pFocusStatus
);
```

### 解説

現在の焦点位置を取得します。プレビュー実行時、CAMSetFocus 関数で焦点位置を設定した後に本関数を実行すると、設定した焦点位置が取得されます。

CAMAutoFocus 関数で焦点を自動調節すると、そのときの焦点位置が取得されます。

### パラメータ

*pFocusStatus*

現在の焦点位置を、1(近接)～10(遠距離)の範囲で取得します。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバがロードされていません
CAM_NOT_STARTED	: プレビューが開始されていません。
CAM_INVALID_PARAM	: 引数が正しくありません
FUNCTION_UNSUPPORTED	: 未サポートエラー



## 4.31 CAMAutoFocus

オートフォーカスによる焦点合わせを行います。

```
[C++]
int CAMAutoFocus (
    DWORD    dwAFMode,
    LPDWORD  pReserved
)
```

```
[Visual Basic]
Public Shared Function CAMAutoFocus ( _
    ByVal dwAFMode As Int32, _
    ByRef pReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMAutoFocus (
    Int32 dwAFMode,
    ref Int32 pReserved
);
```

### 解説

オートフォーカスによる焦点合わせを行います。

本関数を実行すると、ただちに焦点合わせを実行し、終了時に制御を戻します。

### パラメータ

#### *dwAFMode*

オートフォーカスモードを指定します。**CAM\_AF\_NORMAL** を指定してください。

**CAM\_AF\_NORMAL** : 通常はこの値を指定してください

#### *pReserved*

現在のバージョンではこの引数は使用されません。**NULL** を指定してください。

### 戻り値

<b>CAM_SUCCESS</b>	: 正常終了
<b>CAM_NOTOPENED</b>	: カメラドライバがロードされていません
<b>CAM_NOT_STARTED</b>	: プレビューが開始されていません。
<b>CAM_INVALID_PARAM</b>	: 引数が正しくありません
<b>CAM_AF_ERROR</b>	: オートフォーカスに失敗しました Device Emulator では発生しません
<b>CAM_INTERNAL_ERROR</b>	: 内部エラーです Device Emulator では発生しません
<b>FUNCTION_UNSUPPORTED</b>	: 未サポートエラー

## 4.32 CAMSetPreviewMode

プレビューモードを設定します。

```
[C++]
int CAMSetPreviewMode (
    DWORD    dwPreviewMode,
    DWORD    dwReserved
)
```

```
[Visual Basic]
Public Shared Function CAMSetPreviewMode ( _
    ByVal dwPreviewMode As Int32, _
    ByVal dwReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetPreviewMode (
    Int32 dwPreviewMode,
    Int32 dwReserved
);
```

### 解説

プレビューモード(オートフォーカスのフォーカスフレーム表示の有無)を設定します。  
Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、  
CAMGetPreviewMode 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *dwPreviewMode*

プレビューモードを設定します。

- |                        |  |
|------------------------|--|
| CAM_PREVIEW_NORMAL     | : 通常のプレビューです。<br>フォーカスフレームを表示しません(デフォルト) |
| CAM_PREVIEW_FOCUSFRAME | : プレビューにフォーカスフレームを表示します                  |

#### *dwReserved*

現在のバージョンではこの引数は使用されません。NULL を指定してください。

### 戻り値

- |                      |                                       |
|----------------------|---------------------------------------|
| CAM_SUCCESS          | : 正常終了                                |
| CAM_NOTOPENED        | : カメラドライバがロードされていません                  |
| CAM_INVALID_PARAM    | : 引数が正しくありません                         |
| CAM_INTERNAL_ERROR   | : 内部エラーです<br>Device Emulator では発生しません |
| FUNCTION_UNSUPPORTED | : 未サポートエラー                            |

## 4.33 CAMGetPreviewMode

現在のプレビューモードを取得します。

```
[C++]
int CAMGetPreviewMode (
    LPDWORD pPreviewMode,
    LPDWORD pReserved
)
```

```
[Visual Basic]
Public Shared Function CAMGetPreviewMode ( _
    ByRef pPreviewMode As Int32, _
    ByRef pReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMGetPreviewMode (
    ref Int32 pPreviewMode,
    ref Int32 pReserved
);
```

### 解説

現在のプレビューモード(オートフォーカスのフォーカスフレーム設定)を取得します。

### パラメータ

#### *pPreviewMode*

現在のプレビューモードを取得します。

- |                        |  |
|------------------------|--|
| CAM_PREVIEW_NORMAL     | : 通常のプレビューです。<br>フォーカスフレームを表示しません(デフォルト) |
| CAM_PREVIEW_FOCUSFRAME | : プレビューにフォーカスフレームを表示します                  |

#### *pReserved*

現在のバージョンではこの引数は使用されません。NULL を指定してください。

### 戻り値

- |                      |                                       |
|----------------------|---------------------------------------|
| CAM_SUCCESS          | : 正常終了                                |
| CAM_NOTOPENED        | : カメラドライバがロードされていません                  |
| CAM_INVALID_PARAM    | : 引数が正しくありません                         |
| CAM_INTERNAL_ERROR   | : 内部エラーです<br>Device Emulator では発生しません |
| FUNCTION_UNSUPPORTED | : 未サポートエラー                            |

## 4.34 CAMSetVideoCaptureSettings

取り込む動画ファイルのオプションを設定します。

```
[C++]
int CAMSetVideoCaptureSettings (
    DWORD dwVideoCaptureFormat,
    DWORD dwQuality,
    DWORD dwAudio,
    DWORD dwReserved1,
    DWORD dwReserved2
)
```

```
[Visual Basic]
Public Shared Function CAMSetVideoCaptureSettings ( _
    ByVal dwVideoCaptureFormat As Int32, _
    ByVal dwQuality As Int32, _
    ByVal dwAudio As Int32, _
    ByVal dwReserved1 As Int32, _
    ByVal dwReserved2 As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetVideoCaptureSettings (
    Int32 dwVideoCaptureFormat,
    Int32 dwQuality,
    Int32 dwAudio,
    Int32 dwReserved1,
    Int32 dwReserved2
);
```

### 解説

取り込む動画ファイルのオプションを設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetVideoCaptureSettings 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *dwVideoCaptureFormat*

動画ファイルフォーマットを指定します。

CAM\_VIDEOFORMAT\_WMV : WMV ファイル(デフォルト)

#### *dwQuality*

動画ファイルの画質を指定します。

CAM_VIDEO_LOW	: 画質低、圧縮率高
CAM_VIDEO_NORMAL	: 画質中、圧縮率中(デフォルト)
CAM_VIDEO_HIGH	: 画質高、圧縮率低
CAM_VIDEO_HIGHEST	: 画質最高、圧縮率最低

### *dwAudio*

動画ファイルの音声の付加を指定します。

CAM\_VIDEO\_NOTAPPENDAUDIO : 音声を付加しません(デフォルト)  
CAM\_VIDEO\_APPENDAUDIO : 音声を付加します

### *dwReserved1*

現在のバージョンではこの引数を使用しません。NULL を指定してください。

### *dwReserved2*

現在のバージョンではこの引数を使用しません。NULL を指定してください。

## 戻り値

CAM\_SUCCESS : 正常終了  
CAM\_NOTOPENED : カメラドライバをロードしていません  
CAM\_INVALID\_PARAM : 引数が正しくありません  
FUNCTION\_UNSUPPORTED : 未サポートエラー

## 4.35 CAMGetVideoCaptureSettings

取り込む動画ファイルのオプションを取得します。

```
[C++]
int CAMGetVideoCaptureSettings (
    LPDWORD lpVideoCaptureFormat,
    LPDWORD lpQuality,
    LPDWORD lpAudio,
    LPDWORD lpReserved1,
    LPDWORD lpReserved2
)
```

```
[Visual Basic]
Public Shared Function CAMSetVideoCaptureSettings ( _
    ByRef lpVideoCaptureFormat As Int32, _
    ByRef lpQuality As Int32, _
    ByRef lpAudio As Int32, _
    ByRef lpReserved1 As Int32, _
    ByRef lpReserved2 As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetVideoCaptureSettings (
    ref Int32 lpVideoCaptureFormat,
    ref Int32 lpQuality,
    ref Int32 lpAudio,
    ref Int32 lpReserved1,
    ref Int32 lpReserved2
);
```

### 解説

取り込む動画ファイルのオプションを取得します。

### パラメータ

#### *lpVideoCaptureFormat*

動画ファイルフォーマットを取得します。取得する値は、CAMSetVideoCaptureSettings 関数を参照してください。

#### *lpQuality*

動画ファイルの画質を取得します。取得する値は、CAMSetVideoCaptureSettings 関数を参照してください。

#### *lpAudio*

動画ファイルの音声の付加を取得します。取得する値は、CAMSetVideoCaptureSettings 関数を参照してください。

#### *lpReserved1*

現在のバージョンではこの引数を使用しません。

### *lpReserved2*

現在のバージョンではこの引数を使用しません。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
FUNCTION_UNSupport	: 未サポートエラー

## 4.36 CAMSetPreviewFrameRate

プレビュー画面に出力するフレームレートを設定します。

```
[C++]
int CAMSetPreviewFrameRate (
    DWORD dwStillCaptureMaxFps,
    DWORD dwVideoCaptureMaxFps,
    DWORD dwReserved1,
    DWORD dwReserved2
)
```

```
[Visual Basic]
Public Shared Function CAMSetPreviewFrameRate ( _
    ByVal dwStillCaptureMaxFps As Int32, _
    ByVal dwVideoCaptureMaxFps As Int32, _
    ByVal dwReserved1 As Int32, _
    ByVal dwReserved2 As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMSetPreviewFrameRate (
    Int32 dwStillCaptureMaxFps,
    Int32 dwVideoCaptureMaxFps,
    Int32 dwReserved1,
    Int32 dwReserved2
);
```

### 解説

プレビュー画面に出力するフレームレートを設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、CAMGetPreviewFrameRate 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *dwStillCaptureMaxFps*

静止画キャプチャモード時のプレビューにて、カメラが出力する最大フレームレートを指定します。

CAM_MAXFRAMERATE_LOW	: 最大フレームレート低
CAM_MAXFRAMERATE_NORMAL	: 最大フレームレート中
CAM_MAXFRAMERATE_HIGH	: 最大フレームレート高
CAM_MAXFRAMERATE_HIGHEST	: 最大フレームレート最高(デフォルト)

#### *dwVideoCaptureMaxFps*

動画キャプチャモード時のプレビューにて、カメラが出力する最大フレームレートを指定します。

CAM_MAXFRAMERATE_LOW	: 最大フレームレート低
CAM_MAXFRAMERATE_NORMAL	: 最大フレームレート中(デフォルト)
CAM_MAXFRAMERATE_HIGH	: 最大フレームレート高
CAM_MAXFRAMERATE_HIGHEST	: 最大フレームレート最高



- IT-G500 で DT-5300 と同等のフレームレートを実現する場合は、dwVideoCaptureMaxFps に以下を設定してください。

DT-5300	IT-G500
CAM_MAXFRAMERATE_LOW (約 6fps)	CAM_MAXFRAMERATE_NORMAL (約 7fps)
CAM_MAXFRAMERATE_NORMAL (約 9fps)	CAM_MAXFRAMERATE_HIGH (約 10fps)
CAM_MAXFRAMERATE_HIGH (約 15fps)	CAM_MAXFRAMERATE_HIGHEST (約 20fps)

#### *dwReserved1*

現在のバージョンではこの引数を使用しません。NULL を指定してください。

#### *dwReserved2*

現在のバージョンではこの引数を使用しません。NULL を指定してください。

### 戻り値

以下の値を返します。

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.37 CAMGetPreviewFrameRate

プレビュー画面に出力するフレームレートを取得します。

```
[C++]
int CAMGetPreviewFrameRate (
    LPDWORD IpStillCaptureMaxFps,
    LPDWORD IpVideoCaptureMaxFps,
    LPDWORD IpReserved1,
    LPDWORD IpReserved2
)
```

```
[Visual Basic]
Public Shared Function CAMGetPreviewFrameRate ( _
    ByRef IpStillCaptureMaxFps As Int32, _
    ByRef IpVideoCaptureMaxFps As Int32, _
    ByRef IpReserved1 As Int32, _
    ByRef IpReserved2 As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMGetPreviewFrameRate (
    ref Int32 IpStillCaptureMaxFps,
    ref Int32 IpVideoCaptureMaxFps,
    ref Int32 IpReserved1,
    ref Int32 IpReserved2
);
```

### 解説

プレビュー画面に出力するフレームレートを取得します。

### パラメータ

#### *IpStillCaptureMaxFps*

静止画キャプチャモード時のプレビューにて、カメラが出力する最大フレームレートを取得します。取得する値は `CAMSetPreviewFrameRate` 関数を参照してください。

#### *IpVideoCaptureMaxFps*

動画キャプチャモード時のプレビューにて、カメラが出力する最大フレームレートを取得します。取得する値は `CAMSetPreviewFrameRate` 関数を参照してください。

#### *IpReserved1*

現在のバージョンではこの引数を使用しません。

#### *IpReserved2*

現在のバージョンではこの引数を使用しません。

## 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
FUNCTION_UNSupport	: 未サポートエラー

## 4.38 CAMSetCaptureMode

プレビューにおけるキャプチャモードを設定します。

```
[C++]
int CAMSetCaptureMode (
    int    nResMode,
    DWORD  dwCaptureMode
)
```

```
[Visual Basic]
Public Shared Function CAMSetCaptureMode ( _
    ByVal nResMode As Int32, _
    ByVal dwCaptureMode As Int32
) As Int32
```

```
[C#]
public static Int32 CAMSetCaptureMode (
    Int32 nResMode,
    Int32 dwCaptureMode
);
```

### 解説

プレビューにおける静止画キャプチャモードと動画キャプチャモードを切り替えます。  
静止画撮影を行う場合は、本関数で静止画キャプチャモードに設定してください。  
また、動画撮影を行う場合は、本関数で動画キャプチャモードに設定してください。  
Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、  
CAMGetCaptureMode 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *nResMode*

静止画または動画の画像解像度を指定します。

本パラメータと静止画撮影関数 (CAMSingleCapture または CAMSingleCaptureEx 関数) の画像解像度を同じに設定した場合、静止画撮影速度が向上します。

また、本パラメータと動画撮影関数 (CAMStartVideoCapture 関数) の画像解像度を同じに設定した場合、動画撮影開始までの速度が向上します。

CAM_QUADSXGA	: QUAD-SXGA サイズ	(1944x2592) (IT-G500 のみ)
CAM_QUADXGA	: QUAD-XGA サイズ	(1536x2048) (IT-G500 のみ)
CAM_UXGA	: UXGA サイズ	(1200x1600)
CAM_QUADVGA	: QUAD-VGA サイズ	(960x1280)
CAM_XGA	: XGA サイズ	(768x1024)
CAM_SVGA	: SVGA サイズ	(600x800)
CAM_VGA	: VGA サイズ	(480x640)
CAM_QVGA	: 1/4VGA サイズ	(240x320) (デフォルト)

### *dwCaptureMode*

プレビューにおける撮影モードを指定します。

CAM_STILLCAPTURE_MODE	: 静止画キャプチャモード(デフォルト)
CAM_VIDEOCAPTURE_MODE	: 動画キャプチャモード

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_NOT_PREVIEW_STOPPED	: プレビューを停止してください Device Emulator では発生しません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 注意

IT-G500 では、CAMSetCaptureMode 関数とCAMStartVideoCapture 関数の解像度は、同じ値を指定してください。たとえば、VGA サイズの動画を撮影する場合、この 2 つの関数それぞれの nResMode に CAM\_VGA を指定してください。異なる値を指定すると、CAMStartVideoCapture 関数が CAM\_INVALID\_PARAM を返します。

## 4.39 CAMGetCaptureMode

プレビューにおけるキャプチャモードを取得します。

```
[C++]
int CAMSetCaptureMode (
    int      *pResMode,
    LPDWORD  lpCaptureMode
)
```

```
[Visual Basic]
Public Shared Function CAMSetCaptureMode ( _
    ByRef pResMode As Int32, _
    ByRef lpCaptureMode As Int32
) As Int32
```

```
[C#]
public static Int32 CAMSetCaptureMode (
    ref Int32 pResMode,
    ref Int32 lpCaptureMode
);
```

### 解説

プレビューにおけるキャプチャモードを取得します。

### パラメータ

#### *nResMode*

静止画または動画の画像解像度を取得します。取得する値は、CAMSetCaptureMode 関数を参照してください。

#### *dwCaptureMode*

プレビューにおける撮影モードを取得します。取得する値は、CAMSetCaptureMode 関数を参照してください。

### 戻り値

以下の値を返します。

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 4.40 CAMStartVideoCapture

動画撮影とエンコードを開始します。

```
[C++]
int CAMStartVideoCapture(
    const TCHAR *lpzFileName,
    int          nResSize,
    DWORD       dwReserved
)
```

```
[Visual Basic]
Public Shared Function CAMStartVideoCapture( _
    ByVal lpzFileName As String, _
    ByVal nResSize As Int32, _
    ByVal dwReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMStartVideoCapture(
    string lpzFileName,
    Int32 nResSize,
    Int32 dwReserved
);
```

### 解説

動画撮影とエンコードを開始します。

動画撮影は CAMStopVideoCapture 関数を実行するまで継続します。

動画撮影中に、保存先の空き領域が不十分な場合、プログラムメモリ不足が発生した場合、または内部エラーが発生した場合は、動画撮影を自動停止し、イベントとメッセージを発行します。

#### ■ ウィンドウメッセージ通知

WM\_CAM\_AUTOSTOPVCAM( WM\_USER + 0x568 )のウィンドウメッセージを送信します。

また、ウィンドウメッセージの wParam に完了条件を示す以下の値を格納しています。

CAM_VCAP_DISKSHORTAGE	: 保存先の空き容量が残りわずかです
CAM_VCAP_MEMOERYSHORTAGE	: プログラムメモリ不足が発生しました
CAM_VCAP_INTERNALERROR	: 内部エラーです
Device Emulator では発生しません	

#### ■ イベント通知

自動停止時に発行するイベントは“CAMEventAutoStopVideoCapture”です。

Device Emulator では、パラメータチェックのみを行います。

### パラメータ

*lpzFileName*

保存するファイル名をフルパスで指定します。

"¥My Documents"フォルダに"VideoCapture001.wmv"というファイル名で保存するには、

TEXT("¥¥My Documents¥¥VideoCapture001.wmv ")と指定します。

#### *nResSize*

動画の解像度を指定します。

CAM\_QVGA : 1/4VGA サイズ (240x320)  
CAM\_VGA : VGA サイズ (480x640)

OR 指定で動画の向き(回転方向)を指定することができます。(IT-G500 のみ)

CAM\_ROTATE\_NONE : 回転なし  
CAM\_ROTATE\_90 : 時計回りに 90 度回転  
CAM\_ROTATE\_180 : 時計回りに 180 度回転  
CAM\_ROTATE\_270 : 時計回りに 270 度回転

OR 指定で横長画像撮影を指定することができます。(IT-G500 のみ)

CAM\_LANDSCAPE : 横長画像撮影

※ 横長 VGA および 90 度、270 度回転指定時の VGA の解像度は 636x478 ピクセルとなります。

#### *dwReserved*

現在のバージョンではこの引数を使用しません。

#### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_NOT_STARTED	: プレビューが開始されていません
CAM_NOT_VCAP_MODE	: 動画撮影モードではありません CAMSetCaptureMode 関数を実行し、動画撮影モードに 設定してください
CAM_NOT_VCAP_STOPPED	: 動画撮影を停止してください
CAM_NOT_ENOUGH_MEMORY	: プログラム実行用メモリが足りません Device Emulator では発生しません
CAM_FILE_ERROR	: 動画ファイルを保存できません Device Emulator では発生しません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

#### 補足

本関数は、CAMStartPreview 関数実行後に、使用してください。

#### 注意

- CAMSetVideoCaptureSettings 関数でファイルフォーマット WMV、音声付、および最高画質を指定し、CAMSetPreviewFrameRate 関数で最大フレームレート最高、および解像度 QVGA で 30 秒の動画撮影を行う場合、ストレージの容量 10MB 以上、プログラムメモリの残容量 35MB 以上をあげてください。
- CAMSetCaptureMode 関数と CAMStartVideoCapture 関数の解像度は、同じ値を指定してください。たとえば、VGA サイズの動画を撮影する場合、この 2 つの関数それぞれの第一引数に



CAM\_VGA を指定してください。

異なる値を指定すると、CAMStartVideoCapture 関数がパラメータエラーを返します。

- 画像解像度に VGA を指定して長時間の撮影を行う場合は、フレームレートを「標準」以下に設定することをお勧めします。

画質設定とフレームレートを高くすると、メモリ不足で強制的に撮影が中断されたり、動画ファイル作成時間が非常に長くなったりすることがあります。

- 撮影可能時間の目安(画像解像度が VGA の場合)

画質「中」、フレームレート「標準」	: 1 時間以上
画質「高」、フレームレート「高」	: 1 分
画質「最高」、フレームレート「最高」	: 15 秒

- 撮影終了後の動画ファイル作成時間(次の動画が撮影できるまでの時間)の目安(画像解像度が VGA、1 分撮影の場合)

画質「中」、フレームレート「標準」	: 1 秒以下
画質「高」、フレームレート「高」	: 30 秒
画質「最高」、フレームレート「最高」	: 45 秒

## 4.41 CAMStopVideoCapture

動画撮影を停止します。

```
[C++]
int CAMStopVideoCapture(
    DWORD dwReserved
)
```

```
[Visual Basic]
Public Shared Function CAMStopVideoCapture( _
    ByVal dwReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMStopVideoCapture(
    Int32 dwReserved
);
```

### 解説

動画撮影を停止します。

本関数を実行すると、カメラからの画像取り込みを停止しますが、エンコードは継続しています。

エンコード中に、エンコードが完了した場合、保存先が容量不足になった場合、およびプログラムメモリ不足が発生した場合は、エンコードを自動停止します。

その際に、イベントおよびメッセージを発行します。

#### ■ ウィンドウメッセージ通知

WM\_CAM\_AUTOSTOPVCAM( WM\_USER + 0x568 )のウィンドウメッセージを送信します。

また、ウィンドウメッセージの wParam に完了条件を示す以下の値を格納しています。

CAM_VCAP_FILE_COMPLETED	: エンコードが完了しました
CAM_VCAP_DISKSHORTAGE	: 保存先の空き容量が残りわずかです
CAM_VCAP_MEMORYSHORTAGE	: プログラムメモリ不足が発生しました
CAM_VCAP_INTERNALERROR	: 内部エラーです
Device Emulator では発生しません	

#### ■ イベント通知

自動停止時に発行するイベントは“CAMEventAutoStopVideoCapture”です。

### パラメータ

*dwReserved*

現在のバージョンではこの引数を使用しません。NULL を指定してください。

### 戻り値

以下の値を返します。

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_NOT_STARTED	: プレビューが開始されていません

---

CAM_NOT_VCAP_STARTED	: 動画撮影を開始していません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 補足

本関数は、CAMStartVideoCapture 関数実行後に、使用してください。  
本関数を実行すると、プレビューについても停止します。プレビューを再開する場合は、CAMWaitForVideoEncode 関数にて動画データのエンコード完了を待った後に、CAMStartPreview 関数を実行してください。

## 4.42 CAMWaitForVideoEncode

動画のエンコード完了まで待機します。

```
[C++]
int CAMWaitForVideoEncode (
    DWORD    dwMilliseconds,
    LPDWORD  lpEncodeProgressRate,
    DWORD    dwReserved
)
```

```
[Visual Basic]
Public Shared Function CAMWaitForVideoEncode ( _
    ByVal dwMilliseconds As Int32, _
    ByRef lpEncodeProgressRate As Int32, _
    ByVal dwReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMWaitForVideoEncode (
    Int32    dwMilliseconds,
    ref Int32 lpEncodeProgressRate,
    Int32    dwReserved,
);
```

### 解説

動画のエンコード完了まで待機します。

以下のいずれかが成立すると制御を返します。

- 動画のエンコード完了
- エラー発生(エラー内容は **CAMStartVideoCapture** 関数を参照してください)
- タイムアウト時間経過

### パラメータ

#### *dwMilliseconds*

タイムアウト時間をミリ秒単位で指定します。

タイムアウト時間が経過すると、動画のエンコードが未完了であっても、制御を返します。

0 を指定すると、動画のエンコード状態を確認し、即座に制御を返します。

INFINITE を指定すると、動画データのエンコードが完了、または、エラーが発生するまで待機し続けます。

#### *lpEncodeProgressRate*

動画データのエンコードの進捗率(0%～100%(エンコード完了))を取得します。

#### *dwReserved*

現在のバージョンではこの引数を使用しません。NULL を指定してください。

### 戻り値

CAM\_SUCCESS

: 正常終了

CAM_VIDEOENCODING	: 動画のエンコード中です Device Emulator では発生しません
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_NOT_VCAP_STARTED	: 動画撮影を開始していません
CAM_NOT_ENOUGH_MEMORY	: プログラム実行用メモリが足りません Device Emulator では発生しません
CAM_FILE_ERROR	: 動画ファイルを保存できません Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 補足

本関数は、CAMStopVideoCapture 関数実行後に、使用してください。

## 注意

エンコードの進捗率をポーリングするなどの目的で本関数をループ内で繰り返し呼ぶ場合は、dwMilliseconds を必ず 300 以上に指定してください。

## 4.43 CAMCancelVideoEncode

動画のエンコードを中断します。

```
[C++]
int CAMCancelVideoEncode (
    DWORD dwReserved
)
```

```
[Visual Basic]
Public Shared Function CAMCancelVideoEncode ( _
    ByVal dwReserved As Int32 _
) As Int32
```

```
[C#]
public static Int32 CAMCancelVideoEncode (
    Int32 dwReserved
);
```

### 解説

動画のエンコード処理を中断します。

エンコード処理中に本関数を実行すると、エンコードは直ちに終了し、動画ファイルは作成されません。

### パラメータ

*dwReserved*

現在のバージョンではこの引数を使用しません。NULL を指定してください。

### 戻り値

CAM_SUCCESS	: 正常終了
CAM_NOTOPENED	: カメラドライバをロードしていません
CAM_INVALID_PARAM	: 引数が正しくありません
CAM_NOT_VCAP_STOPPED	: 動画撮影を停止していません
CAM_INTERNAL_ERROR	: 内部エラーです Device Emulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

本関数は、CAMStopVideoCapture 関数実行後に、使用してください。

## 5. プログラミング上の注意点

通常は本体内蔵のカメラアプリケーションを使用して撮影を行います。本ライブラリを使用することにより、ユーザアプリケーションから直接カメラモジュールの制御を行うことができます。なお、カメラライブラリとカメラアプリケーションを同時に使用することはできません。

### 5.1 動作フローについて

カメラライブラリを使用する場合は、まずカメラドライバをロードする必要があります。次にプレビューを開始します。静止画や連写画像の撮影は、必ずプレビュー表示中に行う必要があります。なお、プレビュー表示中に本体の電源をオフすると、オン後(レジューム後)は再びプレビューが続行されます。

#### ■ 静止画撮影時の動作フロー

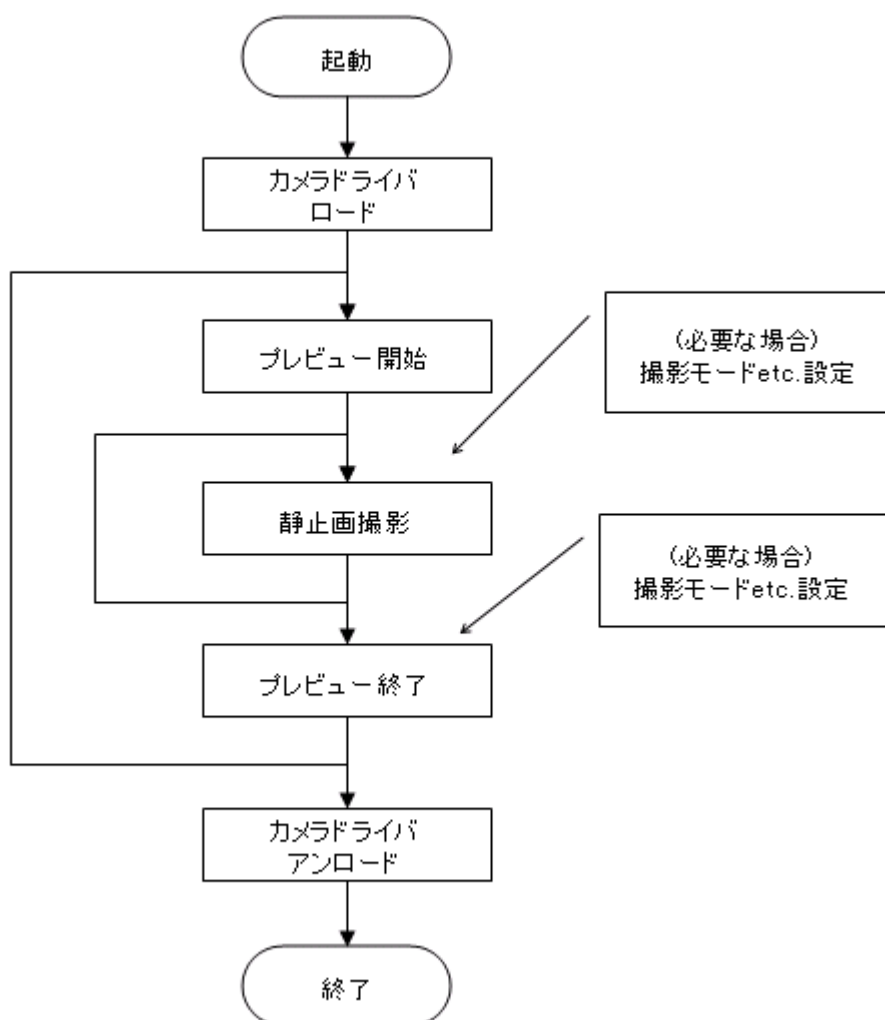


図 5.1 動作フロー

## ■ 動画撮影時の動作フロー

1. **CAMSetVideoCaptureSettings** 関数を実行して、取り込む動画ファイルのオプションを設定します。ここで、音声ありを設定すると、動画撮影と同時に音声録音を実行します。
2. **CAMSetPreviewFrameRate** 関数を実行して、動画キャプチャモード時のプレビューの最大フレームレートを設定します。ここで設定した最大フレームレートは作成する動画ファイルの最大フレームレートと同等になります。
3. **CAMSetCaptureMode** 関数を実行して、プレビューにおけるキャプチャモードを動画キャプチャモードに設定します。
4. **CAMStartPreview** 関数を実行してプレビューを開始します。
5. **CAMStartVideoCapture** 関数を実行して動画撮影を開始します。開始してから **CAMStopVideoCapture** 関数を実行するまでの間に、プログラムメモリ不足や内部エラー等が発生した場合は、動画撮影およびプレビューを自動停止し、**CAM\_WM\_AUTOSTOPVCAP** メッセージまたは名前付きイベント **TEXT("CAMEventAutoStopVideoCapture")** を発行します。アプリケーションプログラムがこれらの自動停止を検知するには、ウィンドウプロシージャや **WaitForSingleObject** 関数等でメッセージもしくはイベントを取得してください。
6. **CAMStopVideoCapture** 関数を実行して動画撮影を停止します。カメラからの画像取り込みを停止しますが、動画データのエンコードは継続します。
7. **CAMWaitForVideoEncode** 関数を実行して動画データのエンコード完了まで待機します。**CAM\_SUCCESS** が返る場合は、動画のファイル作成が完了しています。もう一度撮影する場合は手順 4. のプレビュー開始を実行してください。エンコードを中断する場合は、**CAMCancelVideoEncode** 関数を実行してください。



## 6. Device Emulator

Device Emulator 上で本ライブラリを正しく動作させるためには、以下のファイルが必要です。

CameraLib.inf

\*\*\*.bmp (\*\*\*)は任意)

\*\*\*.jpg (\*\*\*)は任意)

ファイルはあらかじめ以下のフォルダにインストールされています。

¥Storage Card¥Camera

### 6.1 CameraLib.inf

プレビューまたはキャプチャー時に表示する BMP または JPG ファイルのパスを保存するファイルです。  
CameraLib.inf のサンプルを以下に示します。

```
[Preview]
1PER9VGA="¥Storage Card¥Camera¥CAM_1PER9VGA_V. bmp"
QVGA="¥Storage Card¥Camera¥CAM_QVGA_V. bmp"

[Capture]
QVGA_BMP="¥Storage Card¥Camera¥CAM_QVGA. bmp"
QVGA_JPG="¥Storage Card¥Camera¥CAM_QVGA. jpg"
VGA_BMP="¥Storage Card¥Camera¥CAM_VGA. bmp"
VGA_JPG="¥Storage Card¥Camera¥CAM_VGA. jpg"
SVGA_BMP="¥Storage Card¥Camera¥CAM_SVGA. bmp"
SVGA_JPG="¥Storage Card¥Camera¥CAM_SVGA. jpg"
XGA_BMP="¥Storage Card¥Camera¥CAM_XGA. bmp"
XGA_JPG="¥Storage Card¥Camera¥CAM_XGA. jpg"
QUADVGA_BMP="¥Storage Card¥Camera¥CAM_QUADVGA. bmp"
QUADVGA_JPG="¥Storage Card¥Camera¥CAM_QUADVGA. jpg"
UXGA_BMP="¥Storage Card¥Camera¥CAM_UXGA. bmp"
UXGA_JPG="¥Storage Card¥Camera¥CAM_UXGA. jpg"
```

[Preview]    プレビュー画像セクション		
	1PER9VGA	1/9VGA サイズ(160x213)のプレビュー表示用 BMP ファイルのパス
	QVGA	QVGA サイズ(240x320)のプレビュー表示用 BMP ファイルのパス
[Capture]    キャプチャー画像セクション		
	QVGA_BMP	QVGA サイズ(240x320)のキャプチャー表示用 BMP ファイルのパス
	QVGA_JPG	QVGA サイズ(240x320)のキャプチャー表示用 JPG ファイルのパス
	VGA_BMP	VGA サイズ(480x640)のキャプチャー表示用 BMP ファイルのパス
	VGA_JPG	VGA サイズ(480x640)のキャプチャー表示用 JPG ファイルのパス
	SVGA_BMP	SVGA サイズ(600x800)のキャプチャー表示用 BMP ファイルのパス
	SVGA_JPG	SVGA サイズ(600x800)のキャプチャー表示用 JPG ファイルのパス
	XGA_BMP	XGA サイズ(768x1024)のキャプチャー表示用 BMP ファイルのパス
	XGA_JPG	XGA サイズ(768x1024)のキャプチャー表示用 JPG ファイルのパス
	QUADVGA_BMP	QuadVGA サイズ(960x1280)のキャプチャー表示用 BMP ファイルのパス
	QUADVGA_JPG	QuadVGA サイズ(960x1280)のキャプチャー表示用 JPG ファイルのパス
	UXGA_BMP	UXGA サイズ(1200x1600)のキャプチャー表示用 BMP ファイルのパス
	UXGA_JPG	UXGA サイズ(1200x1600)のキャプチャー表示用 JPG ファイルのパス

## 6.2 \*\*\*.bmp

プレビューまたはキャプチャー時に表示する BMP ファイルです。

## 6.3 \*\*\*.jpg

プレビューまたはキャプチャー時に表示する JPG ファイルです。

## カシオ計算機お問い合わせ窓口

### 製品に関する最新情報

- 製品サポートサイト（カシオペア・ハンディターミナル）

<http://casio.jp/support/ht/>

## カシオ計算機株式会社

〒151-8543 東京都渋谷区本町 1-6-2

TEL 03-5334-4638(代)