

Revision	1.70
Date	Jan. 2024

MvUsbCam ライブラリマニュアル

Index

.....	1
[1] 動作環境	1
[2] 開発環境	1
[3] 構造体の説明	2
3.1 MVUSBCAM_SET_GPIO_ALL_INFO... GPIO 制御用構造体	2
3.2 MVUSBCAM_ParamInfo... カメラパラメータ情報構造体	2
3.3 MVUSBCAM_FrameRateInfo... フレームレート情報構造体	3
3.4 MVUSBCAM_GPIO_INFO ... GPIO (有効無効) 情報構造体	3
3.5 MVUSBCAM_UserData_DAT ... ユーザーデータ設定用構造体	3
3.6 MVUSBCAM_WB_ColorGain_PARA ... ホワイトバランス カラーゲイン設定用構造体	4
[4] 関数一覧	5
[5] 処理フローチャート	8
5. 1 全体シーケンス	8
5. 2 画質初期化処理 (USB 接続後) (MCM320 限定)	10
5. 3 画質復元方法(自動露光 OFF 時) (MCM320 限定)	11
[6] 関数説明	12
1) MvUsbCam_Init	12
2) MvUsbCam_Fini	13
3) MvUsbCam_GetDeviceCount	14
4) MvUsbCam_GetDeviceName	15
5) MvUsbCam_GetDevType	16
6) MvUsbCam_GetFirmwareVersion	17
7) MvUsbCam_GetISPVersion	18
8) MvUsbCam_SetLed	19
9) MvUsbCam_GetLed	20
10) MvUsbCam_GetParamInfo	21
11) MvUsbCam_GetFrameRateInfo	22
12) MvUsbCam_SetBrightness	23
13) MvUsbCam_GetBrightness	24
14) MvUsbCam_SetSaturation	25
15) MvUsbCam_GetSaturation	26
16) MvUsbCam_SetSharpness	27
17) MvUsbCam_GetSharpness	28
18) MvUsbCam_SetGamma	29
19) MvUsbCam_GetGamma	30

20) MvUsbCam_SetContrast.....	31
21) MvUsbCam_GetContrast.....	32
22) MvUsbCam_SetHue.....	33
23) MvUsbCam_GetHue.....	34
24) MvUsbCam_SetDenoise.....	35
25) MvUsbCam_GetDenoise.....	36
26) MvUsbCam_SetFlip.....	37
27) MvUsbCam_GetFlip.....	38
28) MvUsbCam_SetMirror.....	39
29) MvUsbCam_GetMirror.....	40
30) MvUsbCam_SetMonochrome.....	41
31) MvUsbCam_GetMonochrome.....	42
32) MvUsbCam_SetStrobeCtrl.....	43
33) MvUsbCam_GetStrobeCtrl.....	45
34) MvUsbCam_SetStrobeWidth.....	46
35) MvUsbCam_GetStrobeWidth.....	47
36) MvUsbCam_GetStrobeWidthReal.....	48
37) MvUsbCam_SetStrobeDelayWidth.....	49
38) MvUsbCam_GetStrobeDelayWidth.....	50
39) MvUsbCam_GetStrobeDelayWidthReal.....	51
40) MvUsbCam_SetSensorOffset.....	52
41) MvUsbCam_GetSensorOffset.....	53
42) MvUsbCam_SetWhiteBalanceAuto.....	54
43) MvUsbCam_GetWhiteBalanceAuto.....	55
44) MvUsbCam_SetWhiteBalanceRed.....	56
45) MvUsbCam_GetWhiteBalanceRed.....	57
46) MvUsbCam_SetWhiteBalanceGreen.....	58
47) MvUsbCam_GetWhiteBalanceGreen.....	59
48) MvUsbCam_SetWhiteBalanceBlue.....	60
49) MvUsbCam_GetWhiteBalanceBlue.....	61
50) MvUsbCam_SetWhiteBalanceRGB.....	62
51) MvUsbCam_GetWhiteBalanceRGB.....	63
52) MvUsbCam_WhiteBalanceUpdate.....	64
53) MvUsbCam_SetWhitePointX.....	65
54) MvUsbCam_GetWhitePointX.....	66
55) MvUsbCam_GetWhitePointX_now.....	67
56) MvUsbCam_SetWhitePointY.....	68
57) MvUsbCam_GetWhitePointY.....	69
58) MvUsbCam_GetWhitePointY_now.....	70
59) MvUsbCam_SetWhitePointXOffset.....	71
60) MvUsbCam_GetWhitePointXOffset.....	72
61) MvUsbCam_GetWhitePointXOffset_now.....	73
62) MvUsbCam_SetWhitePointYOffset.....	74
63) MvUsbCam_GetWhitePointYOffset.....	75
64) MvUsbCam_GetWhitePointYOffset_now.....	76
65) MvUsbCam_SetColorTemperatureEnable.....	77
66) MvUsbCam_GetColorTemperatureEnable.....	78

67) MvUsbCam_SetColorTemperatureValue	79
68) MvUsbCam_GetColorTemperatureValue	80
69) MvUsbCam_SetWBColorConfig_each	81
70) MvUsbCam_GetWBColorConfig_each	82
71) MvUsbCam_SetWBColorConfig_all	83
72) MvUsbCam_GetWBColorConfig_all	84
73) MvUsbCam_SetExposureAuto	85
74) MvUsbCam_GetExposureAuto	86
75) MvUsbCam_SetAeTarget	87
76) MvUsbCam_GetAeTarget	88
77) MvUsbCam_SetFrameRateFix	89
78) MvUsbCam_GetFrameRateFix	90
79) MvUsbCam_SetAnalogGainAuto	91
80) MvUsbCam_GetAnalogGainAuto	92
81) MvUsbCam_SetShutterSpeed	93
82) MvUsbCam_GetShutterSpeed	94
83) MvUsbCam_GetShutterSpeedReal	95
84) MvUsbCam_SetDigitalGain	96
85) MvUsbCam_GetDigitalGain	97
86) MvUsbCam_SetAnalogGain	98
87) MvUsbCam_GetAnalogGain	99
88) MvUsbCam_CameraBusyCheck	100
89) MvUsbCam_CameraBusyWait	101
90) MvUsbCam_CameraBusyWaitCancel	102
91) MvUsbCam_SetShutterMaxMin	103
92) MvUsbCam_GetShutterMaxMin	104
93) MvUsbCam_SetAnaGainMaxMin	105
94) MvUsbCam_GetAnaGainMaxMin	106
95) MvUsbCam_SetDigGainMaxMin	107
96) MvUsbCam_GetDigGainMaxMin	108
97) MvUsbCam_SetFrameRateCtrl	109
98) MvUsbCam_GetFrameRateCtrl	110
99) MvUsbCam_SetFrameRateCommon	111
100) MvUsbCam_GetFrameRateCommon	112
101) MvUsbCam_DefaultGains	113
102) MvUsbCam_SaveGains	114
103) MvUsbCam_SensorInit	115
104) MvUsbCam_GetSensorInitCount	116
105) MvUsbCam_SetGpio	117
106) MvUsbCam_GetGpio	118
107) MvUsbCam_SetGpioAll	119
108) MvUsbCam_GetGpioAll	120
109) MvUsbCam_SetGpioSetting	121
110) MvUsbCam_GetGpioSetting	122
111) MvUsbCam_SetTriggerSetting	123
112) MvUsbCam_GetTriggerSetting	124
113) MvUsbCam_SoftTriggerON	125

114) MvUsbCam_GetGpioInfo	126
115) MvUsbCam_SetGpioPath.....	127
116) MvUsbCam_GetGpioPath	128
117) MvUsbCam_WriteUserData	129
118) MvUsbCam_ReadUserData	130
119) MvUsbCam_StartUserTrgMonitor	131
120) MvUsbCam_StopUserTrgMonitor	132
121) MvUsbCam_GetUserTrgData.....	133
122) MvUsbCam_ClearUserTrgData.....	134
123) MvUsbCam_SetUserTrgEdge	135
124) MvUsbCam_GetUserTrgEdge.....	136
125) MvUsbCam_SetUserTrgChangeCB	137
126) MvUsbCam_SetUserTrgMode	138
127) MvUsbCam_GetUserTrgMode	139
128) MvUsbCam_SetUserTrgLoopTime	140
129) MvUsbCam_GetUserTrgLoopTime	141
130) MvUsbCam_SetPwmSetting.....	142
131) MvUsbCam_GetPwmSetting	143
132) MvUsbCam_PwmStart.....	144
133) MvUsbCam_PwmStop.....	145
134) MvUsbCam_ErrMsg.....	146
135) MvUsbCam_GetErrText.....	147

[1] 動作環境

PC 本体	: PC/AT 互換機
CPU	: Pentium 4 1.4GHz 以上 (推奨 Pentium 4 2.0AGHz 以上)
メモリ	: 256MBytes 以上 (推奨 512MBytes 以上)
USB2.0 HOST	: Intel製 or NEC製
USB2.0 ドライバー	: Microsoft 社製のみ (他社 USB2.0 ドライバーは不可)
USB3.0 HOST	: Intel製 or NEC製
USB3.0 ドライバー	: Microsoft 社製のみ (他社 USB2.0 ドライバーは不可)
OS	: Microsoft Windows 7 (32bit または、64bit) : Microsoft Windows 8.1 (32bit または、64bit) : Microsoft Windows 10 (32bit または、64bit) ※いずれも日本語版

[2] 開発環境

●コンパイラ	Microsoft Visual Studio 2022 C/C++
●ライブラリファイル(32bitOS 用)	
MvUsbCam ライブラリ	……MvUsbCam.lib
上記ライブラリのヘッダファイル	……MvUsbCam.h
上記ライブラリの DLL ファイル	……MvUsbCam.dll
●ライブラリファイル(64bitOS 用)	
MvUsbCam ライブラリ	……MvUsbCam_x64.lib
上記ライブラリのヘッダファイル	……MvUsbCam.h
上記ライブラリの DLL ファイル	……MvUsbCam_x64.dll

※本ライブラリは他社製のカメラには対応しておりません。
ガゾウ社製カメラのみの対応となっております。

[3] 構造体の説明

3.1 MVUSBCAM_SET_GPIO_ALL_INFO... GPIO 制御用構造体

```
typedef struct
{
    DWORD dwVersion;    // [IN] struct version
    BOOL  bGpio1Enb;    // [IN] 1=access, 0=no access
    BOOL  bGpio2Enb;    // [IN] 1=access, 0=no access
    BOOL  bGpio3Enb;    // [IN] 1=access, 0=no access
    BOOL  bGpio4Enb;    // [IN] 1=access, 0=no access
    BOOL  bGpio5Enb;    // [IN] 1=access, 0=no access
    BOOL  bGpio6Enb;    // [IN] 1=access, 0=no access
    BOOL  bTrgEnb;      // [IN] 1=access, 0=no access
    BOOL  bGpio1Dat;    // [IN or OUT] 1=high, 0=low
    BOOL  bGpio2Dat;    // [IN or OUT] 1=high, 0=low
    BOOL  bGpio3Dat;    // [IN or OUT] 1=high, 0=low
    BOOL  bGpio4Dat;    // [IN or OUT] 1=high, 0=low
    BOOL  bGpio5Dat;    // [IN or OUT] 1=high, 0=low
    BOOL  bGpio6Dat;    // [IN or OUT] 1=high, 0=low
    BOOL  bTrgDat;      // [IN or OUT] 1=high, 0=low
} MVUSBCAM_SET_GPIO_ALL_INFO, *PMVUSBCAM_SET_GPIO_ALL_INFO;
```

3.2 MVUSBCAM_ParamInfo... カメラパラメータ情報構造体

```
typedef struct {
    DWORD dwVersion;    // [IN] struct version
    BOOL  bEnable;      // [OUT] 1 = 使用可能, 0 = 不可
    BOOL  bCondi_AWB;    // [OUT] 1 = WhiteBalanceAuto-ON/OFF で、制御条件変化
    BOOL  bEnb_AWB_ON;   // [OUT] 1=WhiteBalanceAuto-ON 時,有効
    BOOL  bEnb_AWB_OFF;  // [OUT] 1 = WhiteBalanceAuto-OFF 時,有効
    BOOL  bCondi_AEC;    // [OUT] 1 = ExposureAuto 状態で、制御条件変化
    BOOL  bEnb_AEC_ON;   // [OUT] 1 = ExposureAuto-ON 時有効
    BOOL  bEnb_AEC_OFF;  // [OUT] 1 = ExposureAuto-OFF 時有効
    BOOL  bCondi_AGC;    // [OUT] 1 = GainAuto 状態で、制御条件変化
    BOOL  bEnb_AGC_ON;   // [OUT] 1 = GainAuto-ON 時有効
    BOOL  bEnb_AGC_OFF;  // [OUT] 1 = GainAuto-OFF 時有効
    DWORD dwMax;        // [OUT] 最大値
    DWORD dwMin;        // [OUT] 最小値
    DWORD dwDef;        // [OUT] デフォルト値
    DWORD dwRes;        // [OUT] 分解能 (Resolution) ... ver1.50 より追加。
    BOOL  bNameEnb;      // [OUT] 1 = label name あり
    char szName[32];     // [OUT] label name
} MVUSBCAM_ParamInfo, *PMVUSBCAM_ParamInfo;
```

3.3 MVUSBCAM_FrameRateInfo... フレームレート情報構造体

```
typedef struct
{
    DWORD dwVersion;    // [IN]   struct version
    int    iMode;        // [IN]   FSP モード (MCM320_FrameRateCtrl_A 等)
    int    iNo;          // [IN]   FSP 序数 (MCM320_FrameRateCtrl_A_Fps15 等)
    BOOL   bEnable;      // [OUT]  TRUE: 有効, FALSE: 無効
    int    iMax;         // [OUT]  FSP 序数の数
    char   szName[32];   // [OUT]  FSP 名
    int    iNameLen;     // [OUT]  FSP 名の文字列数
} MVUSBCAM_FrameRateInfo, *PMVUSBCAM_FrameRateInfo;
```

3.4 MVUSBCAM_GPIO_INFO ... GPIO (有効無効) 情報構造体

```
typedef struct
{
    DWORD dwVersion;    // [IN]   struct version
    BOOL   bGpio1Enb;    // [OUT]  1=access, 0=no access
    BOOL   bGpio2Enb;    // [OUT]  1=access, 0=no access
    BOOL   bGpio3Enb;    // [OUT]  1=access, 0=no access
    BOOL   bGpio4Enb;    // [OUT]  1=access, 0=no access
    BOOL   bGpio5Enb;    // [OUT]  1=access, 0=no access
    BOOL   bGpio6Enb;    // [OUT]  1=access, 0=no access
    BOOL   bTrgInEnb;    // [OUT]  1=access, 0=no access
    BOOL   bTrgOutEnb;   // [OUT]  1=access, 0=no access
    BOOL   bResetEnb;    // [OUT]  1=access, 0=no access
    BOOL   bStandbyEnb;  // [OUT]  1=access, 0=no access
} MVUSBCAM_GPIO_INFO, *PMVUSBCAM_GPIO_INFO;
```

3.5 MVUSBCAM_UserData_DAT ... ユーザーデータ設定用構造体

```
typedef struct
{
    DWORD dwVersion;    // [IN]   struct version
    BYTE  byDat[32];     // [IN/OUT] UserData
} MVUSBCAM_UserData_DAT, *PMVUSBCAM_UserData_DAT;
```

3.6 MVUSBCAM_WB_ColorGain_PARA ... ホワイトバランス カラーゲイン設定用構造体

```
typedef struct
{
    WORD wArea1Gain;    // Area1 Gain, 固定小数 Q8 フォーマット cf. 0x0105 ( Gain = 1.02)
    WORD wArea1Hue;     // Area1 Hue, 固定小数 Q4 フォーマット cf. 0x01E8 ( Hue = 30.5)
    WORD wArea2Gain;    // Area1 Gain 固定小数 Q8 フォーマット cf. 0x0180 ( Gain = 1.5)
    WORD wArea2Hue;     // Area1 Hue 固定小数 Q4 フォーマット cf. 0xFE18 ( Hue = -30.5)
    WORD wArea3Gain;    // Area1 Gain
    WORD wArea3Hue;     // Area1 Hue
    WORD wArea4Gain;    // Area1 Gain
    WORD wArea4Hue;     // Area1 Hue
    WORD wArea5Gain;    // Area1 Gain
    WORD wArea5Hue;     // Area1 Hue
    WORD wArea6Gain;    // Area1 Gain
    WORD wArea6Hue;     // Area1 Hue
    WORD wHueRelative;  // for test
} MVUSBCAM_WB_ColorGain_PARA;
```

[4] 関数一覧

1) MvUsbCam_Init	MvUsbCam ライブラリの初期化处理
2) MvUsbCam_Fini	MvUsbCam ライブラリの終了処理
3) MvUsbCam_GetDeviceCount	MvUsbCam 対応デバイス (カメラ) の数を取得
4) MvUsbCam_GetDeviceName	デバイス名を取得
5) MvUsbCam_GetDevType	デバイスタイプの取得
6) MvUsbCam_GetFirmwareVersion	ファームウェアバージョンの取得
7) MvUsbCam_GetISPVersion	ISP バージョンの取得
8) MvUsbCam_SetLed	L E D点灯制御
9) MvUsbCam_GetLed	L E D点灯状態の取得
10) MvUsbCam_GetParamInfo	パラメータ情報の取得
11) MvUsbCam_GetFrameRateInfo	フレームレート情報の取得
12) MvUsbCam_SetBrightness	明るさ (ブライトネス値) の設定
13) MvUsbCam_GetBrightness	明るさ (ブライトネス値) の取得
14) MvUsbCam_SetSaturation	彩度 (サチュレーション値) の設定
15) MvUsbCam_GetSaturation	彩度 (サチュレーション値) の取得
16) MvUsbCam_SetSharpness	鮮明度 (シャープネス値) の設定
17) MvUsbCam_GetSharpness	鮮明度 (シャープネス値) の取得
18) MvUsbCam_SetGamma	ガンマ値の設定
19) MvUsbCam_GetGamma	ガンマ値の取得
20) MvUsbCam_SetContrast	コントラスト値の設定
21) MvUsbCam_GetContrast	コントラスト値の取得
22) MvUsbCam_SetHue	Hue 値の設定
23) MvUsbCam_GetHue	Hue 値の取得
24) MvUsbCam_SetDenoise	デノイズの設定
25) MvUsbCam_GetDenoise	デノイズの取得
26) MvUsbCam_SetFlip	上下反転の設定
27) MvUsbCam_GetFlip	上下反転状態の取得
28) MvUsbCam_SetMirror	左右反転の設定
29) MvUsbCam_GetMirror	左右反転状態の取得
30) MvUsbCam_SetMonochrome	白黒設定
31) MvUsbCam_GetMonochrome	白黒状態の取得
32) MvUsbCam_SetStrobeCtrl	ストロボ信号の設定
33) MvUsbCam_GetStrobeCtrl	ストロボ信号の取得
34) MvUsbCam_SetStrobeWidth	ストロボ幅の設定
35) MvUsbCam_GetStrobeWidth	ストロボ幅の取得
36) MvUsbCam_GetStrobeWidthReal	ストロボ幅 (実時間) の取得
37) MvUsbCam_SetStrobeDelayWidth	ストロボディレイ幅の設定
38) MvUsbCam_GetStrobeDelayWidth	ストロボディレイ幅の取得
39) MvUsbCam_GetStrobeDelayWidthReal	ストロボディレイ幅 (実時間) の取得
40) MvUsbCam_SetSensorOffset	センサオフセット値の設定
41) MvUsbCam_GetSensorOffset	センサオフセット値の取得
42) MvUsbCam_SetWhiteBalanceAuto	自動ホワイトバランスの設定
43) MvUsbCam_GetWhiteBalanceAuto	自動ホワイトバランス値の設定
44) MvUsbCam_SetWhiteBalanceRed	ホワイトバランス 赤レベルの設定

45) MvUsbCam_GetWhiteBalanceRed	ホワイトバランス 赤レベルの取得
46) MvUsbCam_SetWhiteBalanceGreen	ホワイトバランス 緑レベルの設定
47) MvUsbCam_GetWhiteBalanceGreen	ホワイトバランス 緑レベルの取得
48) MvUsbCam_SetWhiteBalanceBlue	ホワイトバランス 青レベルの設定
49) MvUsbCam_GetWhiteBalanceBlue	ホワイトバランス 青レベルの取得
50) MvUsbCam_SetWhiteBalanceRGB	ホワイトバランス 赤緑青レベルの一括設定
51) MvUsbCam_GetWhiteBalanceRGB	ホワイトバランス 赤緑青レベルの一括取得
52) MvUsbCam_WhiteBalanceUpdate	ホワイトバランス更新
53) MvUsbCam_SetWhitePointX	ホワイトポイントX値の設定
54) MvUsbCam_GetWhitePointX	ホワイトポイントX値の取得
55) MvUsbCam_GetWhitePointX_now	ホワイトポイントX現在値の取得
56) MvUsbCam_SetWhitePointY	ホワイトポイントY値の設定
57) MvUsbCam_GetWhitePointY	ホワイトポイントY値の取得
58) MvUsbCam_GetWhitePointY_now	ホワイトポイントY現在値の取得
59) MvUsbCam_SetWhitePointXOffset	ホワイトポイントオフセットX値の設定
60) MvUsbCam_GetWhitePointXOffset	ホワイトポイントオフセットX値の取得
61) MvUsbCam_GetWhitePointXOffset_now	ホワイトポイントオフセットX現在値の取得
62) MvUsbCam_SetWhitePointYOffset	ホワイトポイントオフセットY値の設定
63) MvUsbCam_GetWhitePointYOffset	ホワイトポイントオフセットY値の取得
64) MvUsbCam_GetWhitePointYOffset_now	ホワイトポイントオフセットY現在値の取得
65) MvUsbCam_SetColorTemperatureEnable	色温度の有効無効値の設定
66) MvUsbCam_GetColorTemperatureEnable	色温度の有効無効値の取得
67) MvUsbCam_SetColorTemperatureValue	色温度値の設定
68) MvUsbCam_GetColorTemperatureValue	色温度値の取得
69) MvUsbCam_SetWBColorConfig_each	カラーバランス調整値の設定 (エリア指定)
70) MvUsbCam_GetWBColorConfig_each	カラーバランス調整値の取得 (エリア指定)
71) MvUsbCam_SetWBColorConfig_all	カラーバランス調整値の設定 (全エリア)
72) MvUsbCam_GetWBColorConfig_al	カラーバランス調整値の取得 (全エリア)
73) MvUsbCam_SetExposureAuto	自動露光の設定
74) MvUsbCam_GetExposureAuto	自動露光値の取得
75) MvUsbCam_SetAeTarget	自動露光ターゲット値の設定
76) MvUsbCam_GetAeTarget	自動露光ターゲット値の取得
77) MvUsbCam_SetFrameRateFix	フレームレート固定の設定
78) MvUsbCam_GetFrameRateFix	フレームレート固定の取得
79) MvUsbCam_SetAnalogGainAuto	アナログゲインオートの設定
80) MvUsbCam_GetAnalogGainAuto	アナログゲインオート状態の取得
81) MvUsbCam_SetShutterSpeed	シャッタースピードの設定
82) MvUsbCam_GetShutterSpeed	シャッタースピードの取得
83) MvUsbCam_GetShutterSpeedReal	シャッタースピード(実時間) の取得
84) MvUsbCam_SetDigitalGain	デジタルゲイン設定
85) MvUsbCam_GetDigitalGain	デジタルゲイン設定の取得
86) MvUsbCam_SetAnalogGain	アナログゲイン設定
87) MvUsbCam_GetAnalogGain	アナログゲイン値の取得
88) MvUsbCam_CameraBusyCheck	カメラビジーチェック
89) MvUsbCam_CameraBusyWait	カメラビジー (の終了) 待ち
90) MvUsbCam_CameraBusyWaitCancel	MvUsbCam_CameraBusyWait のキャンセル
91) MvUsbCam_SetShutterMaxMin	シャッタースピード上限値／下限値の設定

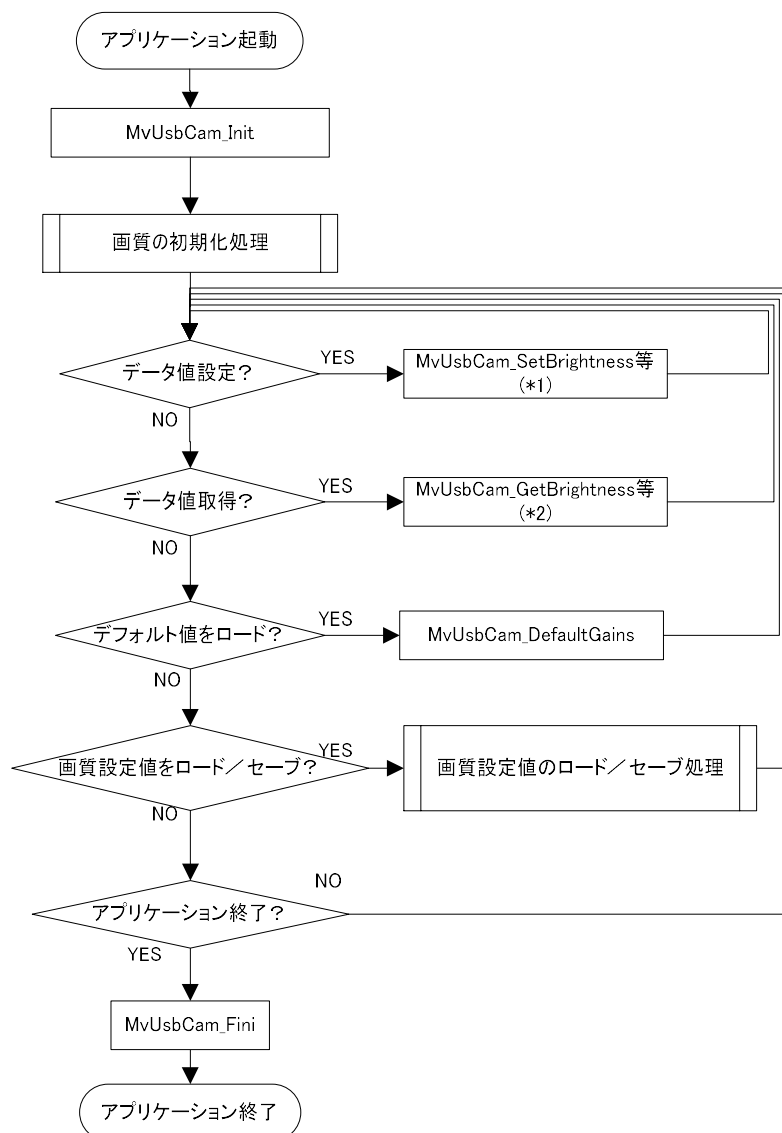
92) MvUsbCam_GetShutterMaxMin	シャッタースピード上限値／下限値の取得
93) MvUsbCam_SetAnaGainMaxMin	アナログゲイン上限値／下限値の設定
94) MvUsbCam_GetAnaGainMaxMin	アナログゲイン上限値／下限値の取得
95) MvUsbCam_SetDigGainMaxMin	デジタルゲイン上限値／下限値の設定
96) MvUsbCam_GetDigGainMaxMin	デジタルゲイン上限値／下限値の取得
97) MvUsbCam_SetFrameRateCtrl	フレームレートの設定
98) MvUsbCam_GetFrameRateCtrl	フレームレート値の取得
99) MvUsbCam_SetFrameRateCommon	共通フレームレート値の設定
100) MvUsbCam_GetFrameRateCommon	共通フレームレート値の取得
101) MvUsbCam_DefaultGains	画質調整値群の初期化
102) MvUsbCam_SaveGains	画質調整値群のセーブ
103) MvUsbCam_SensorInit	センサー初期化
104) MvUsbCam_GetSensorInitCount	センサー初期化回数の取得
105) MvUsbCam_SetGpio	GPIO の出力
106) MvUsbCam_GetGpio	GPIO 値の取得
107) MvUsbCam_SetGpioAll	GPIO の一括出力
108) MvUsbCam_GetGpioAll	GPIO 値の一括取得
109) MvUsbCam_SetGpioSetting	GPIO 設定
110) MvUsbCam_GetGpioSetting	GPIO 設定値の取得
111) MvUsbCam_SetTriggerSetting	外部トリガー設定
112) MvUsbCam_GetTriggerSetting	外部トリガー値の取得
113) MvUsbCam_SoftTriggerON	SW トリガーON 処理
114) MvUsbCam_GetGpioInfo	GPIO (有効無効) 情報の取得
115) MvUsbCam_SetGpioPath	GPIO PATH の設定
116) MvUsbCam_GetGpioPath	GPIO PATH 値の取得
117) MvUsbCam_WriteUserData	ユーザーデータの書き込み
118) MvUsbCam_ReadUserData	ユーザーデータの読み出し
119) MvUsbCam_StartUserTrgMonitor	ユーザートリガーのモニタ開始
120) MvUsbCam_StopUserTrgMonitor	ユーザートリガーのモニタ停止
121) MvUsbCam_GetUserTrgData	ユーザートリガーのデータ取得
122) MvUsbCam_ClearUserTrgData	ユーザートリガーのデータクリア
123) MvUsbCam_SetUserTrgEdge	ユーザートリガーエッジ指定の設定
124) MvUsbCam_GetUserTrgEdge	ユーザートリガーエッジ指定値の取得
125) MvUsbCam_SetUserTrgChangeCB	ユーザートリガーコールバック関数の設定
126) MvUsbCam_SetUserTrgMode	ユーザートリガーモードの設定
127) MvUsbCam_GetUserTrgMode	ユーザートリガーモード値の取得
128) MvUsbCam_SetUserTrgLoopTime	ユーザートリガーループ時間の設定
129) MvUsbCam_GetUserTrgLoopTime	ユーザートリガーループ時間の取得
130) MvUsbCam_SetPwmSetting	PWM の設定
131) MvUsbCam_GetPwmSetting	PWM 値の取得
132) MvUsbCam_PwmStart	PWM スタート
133) MvUsbCam_PwmStop	PWM ストップ
134) MvUsbCam_ErrMsg	エラーメッセージの表示
135) MvUsbCam_GetErrMsg	エラーメッセージの取得

[5] 処理フローチャート

以下、処理フローチャートの一覧です。

- 5. 1 全体シーケンス
- 5. 2 画質初期化处理 (USB 接続後)
- 5. 3 画質復元处理 (自動露光OFF時)

5. 1 全体シーケンス



*1 ...画質調整等のデータ値設定関数として、以下のような関数があります。

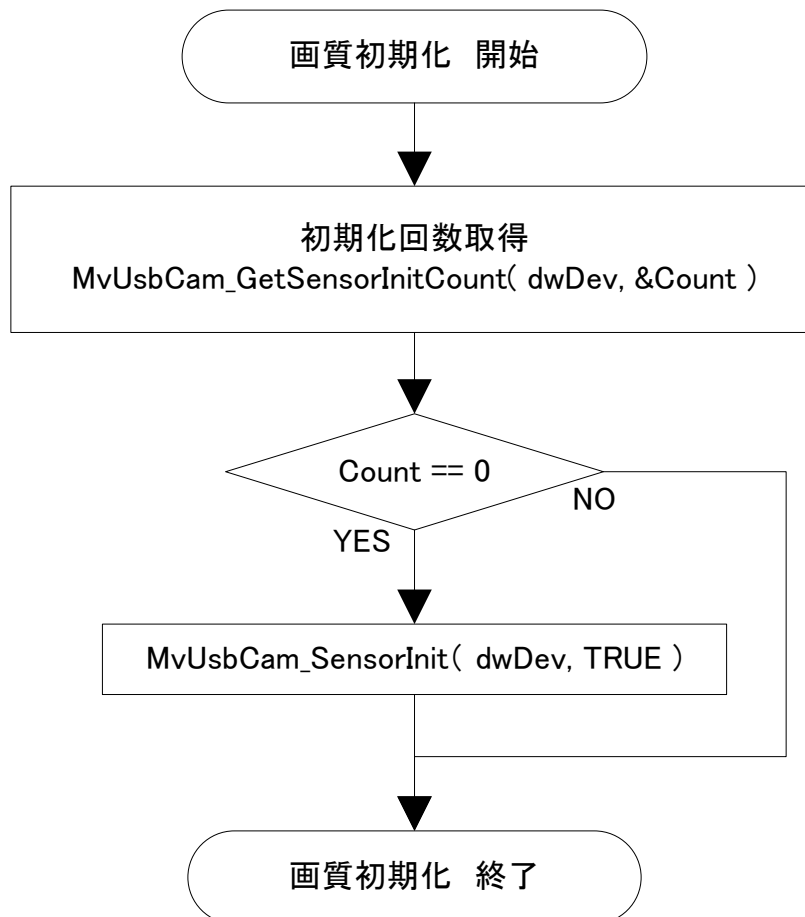
MvUsbCam_SetLed	MvUsbCam_SetBrightness
MvUsbCam_SetSaturation	MvUsbCam_SetSharpness
MvUsbCam_SetGamma	MvUsbCam_SetContrast
MvUsbCam_SetFlip	MvUsbCam_SetMirror
MvUsbCam_SetMonochrome	MvUsbCam_SetWhiteBalanceAuto
MvUsbCam_SetWhiteBalanceRed	MvUsbCam_SetWhiteBalanceGreen
MvUsbCam_SetWhiteBalanceBlue	MvUsbCam_SetWhiteBalanceRGB
MvUsbCam_SetExposureAuto	MvUsbCam_SetFrameRateFix
MvUsbCam_SetShutterSpeed	MvUsbCam_SetDigitalGain
MvUsbCam_SetAnalogGain	MvUsbCam_SetFrameRateCtrl
MvUsbCam_WhiteBalanceUpdate	

*2 ...画質調整等のデータ値取得関数として、以下のような関数があります。

MvUsbCam_GetLed	MvUsbCam_GetBrightness
MvUsbCam_GetSaturation	MvUsbCam_GetSharpness
MvUsbCam_GetGamma	MvUsbCam_GetContrast
MvUsbCam_GetFlip	MvUsbCam_GetMirror
MvUsbCam_GetMonochrome	MvUsbCam_GetWhiteBalanceAuto
MvUsbCam_GetWhiteBalanceRed	MvUsbCam_GetWhiteBalanceGreen
MvUsbCam_GetWhiteBalanceBlue	MvUsbCam_GetWhiteBalanceRGB
MvUsbCam_GetExposureAuto	MvUsbCam_GetFrameRateFix
MvUsbCam_GetShutterSpeed	MvUsbCam_GetDigitalGain
MvUsbCam_GetAnalogGain	MvUsbCam_GetFrameRateCtrl

5. 2 画質初期化処理（USB 接続後）（MCM320 限定）

USB 接続後、少なくとも一回だけ、MvUsbCam_SensorInit を C A L L していただく必要があります。以下は、USB 接続後、一回だけ、MvUsbCam_SensorInit を C A L L するための処理シーケンスです。（本処理シーケンスは、**MCM320** にのみ必要です。）

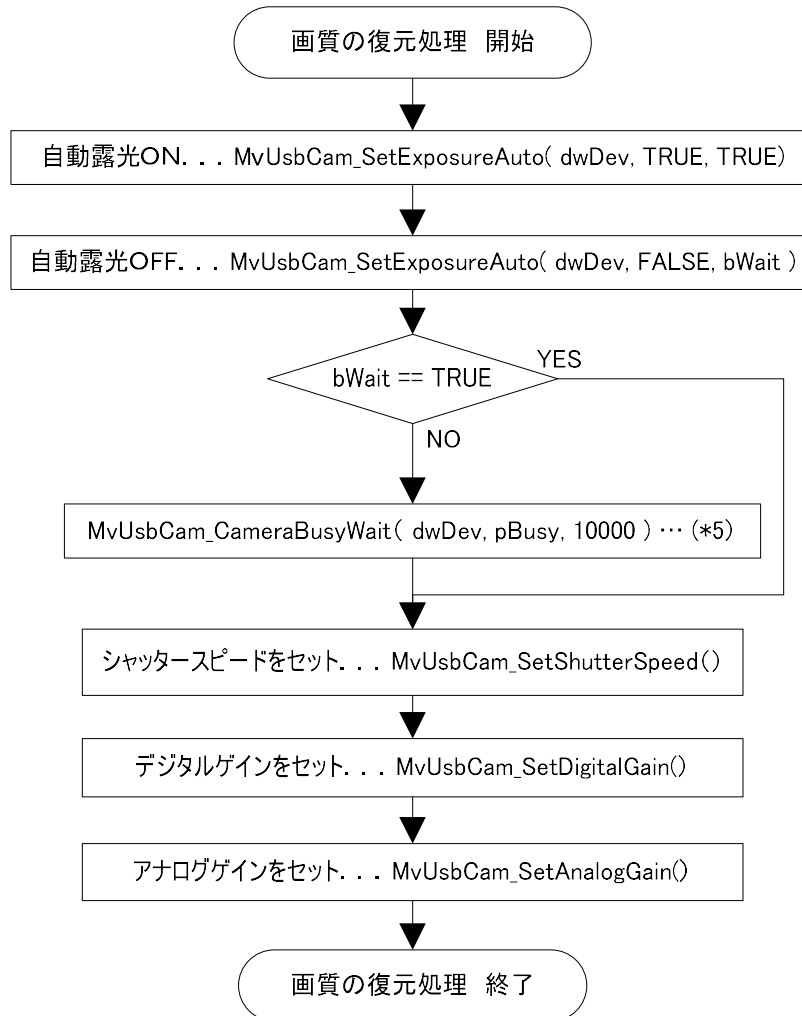


5. 3 画質復元方法(自動露光OFF時) (MCM320 限定)

自動露光をOFFする時や、「マニュアル露光(MvUsbCam_SetExposureAuto の第2引数を FALSE にした)状態で」、USB 再接続や、Windows 再起動した場合、以前と異なる画質になる場合があります。

以前の画質に復元するためには、以下の手順を実行してください。

(本処理シーケンスは、MCM320 にのみ必要です。)



* 5 ...MvUsbCam_CameraBusyWait により、カメラが「ビジー状態でなくなるまで待つ処理」です。

「ビジー状態でなくなるまで待つ処理」は、フレームレートが小さいほど、照度が低いほど時間がかかります。待ち時間の目安は以下の通りです。

U X G A サイズの場合は、最大約 1 0 秒。

V G A 以下の場合は、最大約 2 秒。

また、自動露光OFF時に、MvUsbCam_SetExposureAuto の第3引数(bWait)が重要です。

bWait==TRUE の場合は、関数 MvUsbCam_SetExposureAuto 内部で、MvUsbCam_CameraBusyWait を実行していますが、

bWait==FALSE の場合は、関数 MvUsbCam_SetExposureAuto の後に、MvUsbCam_CameraBusyWait を C A L L していただく必要があります。

[6] 関数説明

ライブラリの関数を詳しく説明します。

1) MvUsbCam_Init
 DWORD MvUsbCam_Init(void)

引数

なし

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。
MVUSBCAM_NOT_OPEN_DEVICE : デバイスオープンエラー

説明

MvUsbCam ライブラリの初期化処理を実行します。

2) MvUsbCam_Fini

DWORD MvUsbCam_Fini(void)

引数

なし

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_NOT_OPEN_DEVICE : デバイスオープンエラー

説明

MvUsbCam ライブラリの終了処理を実行します。

3) MvUsbCam_GetDeviceCount

DWORD MvUsbCam_GetDeviceCount(DWORD *pNum)

引数

*pNum MvUsbCam 対応デバイスの数を返します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_NOT_OPEN_DEVICE : デバイスオープンエラー

説明

MvUsbCam 対応デバイスの数を返します。

4) MvUsbCam_GetDeviceName

DWORD MvUsbCam_GetDeviceName(DWORD dwDev, char *pName)

引数

dwDev デバイス番号(0～MVUSBCAM_DEVICE_MAX - 1) ※通常は0
pName デバイス名を返します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_NOT_OPEN_DEVICE : デバイスオープンエラー

説明

MvUsbCam 対応デバイス名を返します。

5) MvUsbCam_GetDevType

DWORD MvUsbCam_GetDevType(DWORD dwDev, int *pType)

引数

dwDev デバイス番号(0 ～ MvUsbCam_DEVICE_MAX - 1) ※通常は 0
 *pType デバイスタイプ
 MVUSBCAM_CAMTYPE_MCM320、
 MVUSBCAM_CAMTYPE_MCM303 等

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_NOT_OPEN_DEVICE : デバイスオープンエラー
 MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います。
 MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー。
 MVUSBCAM_INVALID_PARAM : パラメータが無効です。
 MVUSBCAM_NO_SUPPORTED_CAM : 未サポートのカメラです。

説明

デバイスタイプを取得します。

6) MvUsbCam_GetFirmwareVersion

DWORD MvUsbCam_GetFirmwareVersion(DWORD dwDev, WORD *pDat)

引数

dwDev	デバイス番号(0 ～ MvUsbCam_DEVICE_MAX - 1) ※通常は 0
*pDat	ファームウェアバージョン

*pDat = 0x0101 は、 Ver1.01 を示します。

*pDat = 0x0213 は、 Ver2.13 を示します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_NOT_OPEN_DEVICE	: デバイスオープンエラー
MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います。
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー。
MVUSBCAM_INVALID_PARAM	: パラメータが無効です。
MVUSBCAM_NO_SUPPORTED_CAM	: 未サポートのカメラです。

説明

ファームウェアバージョンを取得します。

7) MvUsbCam_GetISPVersion

DWORD MvUsbCam_GetISPVersion(DWORD dwDev, WORD *pDat)

引数

dwDev デバイス番号(0 ～ MvUsbCam_DEVICE_MAX - 1) ※通常は 0
*pDat ISP ファームウェアバージョン

*pDat = 0x0101 は、 Ver1.01 を示します。

*pDat = 0x0213 は、 Ver2.13 を示します。

また、GZ4120 以外の時は、0x0000 を返します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_NOT_OPEN_DEVICE : デバイスオープンエラー
MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います。
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー。
MVUSBCAM_INVALID_PARAM : パラメータが無効です。
MVUSBCAM_NO_SUPPORTED_CAM : 未サポートのカメラです。

説明

ファームウェアバージョンを取得します。

8) MvUsbCam_SetLed

DWORD MvUsbCam_SetLed(DWORD dwDev, DWORD dwDat)

引数

dwDev デバイス番号(0 ～ MvUsbCam_DEVICE_MAX - 1) ※通常は 0
dwDat L E D の点灯状態(0 : 消灯 1 : 点灯)
 または、LED2 (注 2) の出力状態 (0 = L O W, 1 = H I G H)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_NOT_OPEN_DEVICE : デバイスオープンエラー
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

L E D 点灯制御 (注 1) を行います。

または、センサ基板側の LED2 (注 2) の出力制御を行います。

また、GZ320/GZ303 の場合は、GPOUT2(J2 の 4Pin) (注 3) で出力します。

注 1... オプションの L E D 基板搭載時のみ有効です。

注 2... 取扱説明書 for MCM320.pdf、または、取扱説明書 for MCM303.pdf の
2.4.1 章の J1 を参照してください。

注 3... GPOUT2 は、STROBE 信号と切り替える事ができます。
そのため、MvUsbCam_SetGpioPath(dwDev,1,0)を Call する事で、
GPOUT2 出力が有効にしておく必要があります。

9) MvUsbCam_GetLed

DWORD MvUsbCam_GetLed(DWORD dwDev, DWORD *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 pDat LED の点灯状態(0 : 消灯 1 : 点灯)
 または、LED2 (注 2) の出力状態 (0 = LOW, 1 = HIGH)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_NOT_OPEN_DEVICE : デバイスオープンエラー
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

LED の点灯状態 (注 1) を取得します。

または、センサ基板側の LED2 (注 2) の出力状態を取得します。

また、GZ320/GZ303 の場合は、GPOUT2(J2 の 4Pin) (注 3) の出力状態を取得します。

注 1... オプションの LED 基板搭載時のみ有効です。

注 2... 取扱説明書 for MCM320.pdf、または、取扱説明書 for MCM303.pdf の
 2.4.1 章の J1 を参照してください。

注 3... GPOUT2 は、STROBE 信号と切り替える事ができます。
 そのため、MvUsbCam_SetGpioPath(dwDev,1,0)を Call する事で、
 GPOUT2 出力が有効にしておく必要があります。

10) MvUsbCam_GetParamInfo

DWORD MvUsbCam_GetParamInfo(DWORD dwDev, DWORD dwIndex,
MVUSBCAM_ParamInfo *pProp)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
dwType	パラメータ番号 MVUSBCAM_ParamIndex_BRIGHTNESS 等
pProp	パラメータ情報

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー

説明

パラメータ情報を取得します。

MVUSBCAM_ParamInfo 構造体要素の dwVersion を、
MVUSBCAM_ParamInfo_VERSION に変更してから、CALL してください。

11) MvUsbCam_GetFrameRateInfo

DWORD MvUsbCam_GetFrameRateInfo(DWORD dwDev, MVUSBCAM_FrameRateInfo *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat フレームレート情報

typedef struct

```
{
    DWORD dwVersion;    // [IN]   struct version
    int    iMode;        // [IN]   FSP モード A or B or C (MVUSBCAM_FrameRateCtrl_A 等)
    int    iNo;          // [IN]   FSP 序数      (MCM320_FrameRateCtrl_A_Fps15 等)
    BOOL   bEnable;      // [OUT]  TRUE: 有効, FALSE: 無効
    int    iMax;         // [OUT]  FSP 序数の数
    char    szName[32];  // [OUT]  FSP 名
    int    iNameLen;     // [OUT]  FSP 名の文字列数
} MVUSBCAM_FrameRateInfo, *PMVUSBCAM_FrameRateInfo;
```

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

パラメータ情報を取得します。

上記構造体の[IN]のパラメータを指定して、関数CALLする事で、
[OUT]のパラメータを取得できます。

MVUSBCAM_FrameRateInfo 構造体要素の dwVersion を、
MVUSBCAM_FrameRateInfo_VERSION に変更してから、CALLしてください。

12) MvUsbCam_SetBrightness

DWORD MvUsbCam_SetBrightness(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat 明るさ (ブライトネス値)
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

明るさ (ブライトネス値) を設定します。
本関数 CALL 後に取得した入力画像に反映します。

13) MvUsbCam_GetBrightness

DWORD MvUsbCam_GetBrightness(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat 明るさ (ブライトネス値) を返します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

明るさ (ブライトネス値) を取得します。

14) MvUsbCam_SetSaturation

DWORD MvUsbCam_SetSaturation(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat 彩度 (サチュレーション値)
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

彩度 (サチュレーション値) を設定します。
本関数 CALL 後に取得した入力画像に反映します。

15) MvUsbCam_GetSaturation

DWORD MvUsbCam_GetSaturation(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat 彩度 (サチュレーション値)
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

彩度 (サチュレーション値) を取得します。

16) MvUsbCam_SetSharpness

DWORD MvUsbCam_SetSharpness(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

鮮明度（シャープネス値）を設定します。

本関数 CALL 後に取得した入力画像に反映します。

17) MvUsbCam_GetSharpness

DWORD MvUsbCam_GetSharpness(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

鮮明度（シャープネス値）を取得します。

18) MvUsbCam_SetGamma

DWORD MvUsbCam_SetGamma(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ガンマ値を設定します。

本関数 CALL 後に取得した入力画像に反映します。

19) MvUsbCam_GetGamma

DWORD MvUsbCam_GetGamma(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ガンマ値を取得します。

20) MvUsbCam_SetContrast

DWORD MvUsbCam_SetContrast(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

コントラスト値を設定します。

本関数 CALL 後に取得した入力画像に反映します。

21) MvUsbCam_GetContrast

DWORD MvUsbCam_GetContrast(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

コントラスト値を取得します。

22) MvUsbCam_SetHue

DWORD MvUsbCam_SetContrast(DWORD dwDev, int iDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat	色相値 (ヒュー値)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

色相 (ヒュー値) を設定します。

本関数 CALL 後に取得した入力画像に反映します。

本関数は、GZ4120 専用関数です。

23) MvUsbCam_GetHue

DWORD MvUsbCam_GetHue(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat 色相値 (ヒュー値)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

色相 (ヒュー値) を取得します。
 本関数は、GZ4120 専用関数です。

24) MvUsbCam_SetDenoise

DWORD MvUsbCam_SetDenoise(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat デノイズ値 (ノイズ除去レベル)
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

デノイズ値 (ノイズ除去レベル) を設定します。
本関数 CALL 後に取得した入力画像に反映します。
本関数は、GZ320/GZ303 専用関数です。

25) MvUsbCam_GetDenoise

DWORD MvUsbCam_GetDenoise(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat デノイズ値 (ノイズ除去レベル) を返します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

デノイズ値 (ノイズ除去レベル) を取得します。
本関数は、GZ320/GZ303 専用関数です。

26) MvUsbCam_SetFlip

DWORD MvUsbCam_SetFlip(DWORD dwDev, BOOL bEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bEnb 上下反転設定値(TRUE : あり FALSE : なし)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

上下反転を設定します。
本関数 CALL 後に取得した入力画像に反映します。

27) MvUsbCam_GetFlip

DWORD MvUsbCam_GetFlip(DWORD dwDev, BOOL *pEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pEnb 上下反転設定値(TRUE : あり FALSE : なし)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

上下反転状態を取得します。

28) MvUsbCam_SetMirror

DWORD MvUsbCam_SetMirror(DWORD dwDev, BOOL bEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bEnb 左右反転設定値(1 : あり、 0 : なし)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

左右反転を設定します。
本関数 CALL 後に取得した入力画像に反映します。

29) MvUsbCam_GetMirror

DWORD MvUsbCam_GetMirror(DWORD dwDev, BOOL *pEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pEnb 左右反転設定値(1 : あり、 0 : なし)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

左右反転状態を取得します。

30) MvUsbCam_SetMonochrome

DWORD MvUsbCam_SetMonochrome(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat 白黒状態を設定します。(1 : 白黒、 0 : カラー)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

白黒状態を設定します。

31) MvUsbCam_GetMonochrome

DWORD MvUsbCam_GetMonochrome(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat 白黒状態を設定します。(1 : 白黒、 0 : カラー)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

白黒設定状態を取得します。

32) MvUsbCam_SetStrobeCtrl

DWORD MvUsbCam_SetStrobeCtrl(DWORD dwDev, BOOL bEnb, BOOL bInvert)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bEnb 1=出力する。(0=出力しない。)
bInvert 1=反転出力する。(0=反転しない。)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ストロボ信号を設定します。

ストロボ信号は、MCM303,GZ4120,GZ320,GZ303 でのみ出力できます。

MCM303 は、 LED1 (= センサ基板側 J2 の Pin2)から出力します。

GZ4120 は、 GPO から出力します。

GZ320,GZ303 は、GPOUT2(= J2 の Pin4)から出力します。

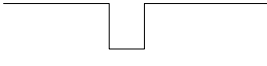
MCM303 の場合

Enb	Invert	ストロボ出力(LED1)
0	0	LOW 固定
0	1	High 固定
1	0	
1	1	

GZ4120 の場合

Enb	Invert	ストロボ出力(GPO)
0	0	LOW 固定
0	1	LOW 固定
1	0	
1	1	

GZ320,GZ303 の場合(*1)

Enb	Invert	ストロボ出力 (GPOUT2)
0	Don't care	HIGH 固定
1	Don't care	

*1 ... ただし、MvUsbCam_SetGpioPath(dwDev,1,1)を Call する事で、
Strobe 出力を有効にしておく必要があります。
しかし、MvUsbCam_SetGpioPath(dwDev,1,0)を Call する事で、
GPOUT2 出力が有効になっている場合は、
MvUsbCam_SetGpio(dwDev, 1, iDat) でのみ操作が可能になります。

33) MvUsbCam_GetStrobeCtrl

DWORD MvUsbCam_GetStrobeCtrl(DWORD dwDev, BOOL *pEnb, BOOL *pInvert)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pEnb 1=出力する。(0=出力しない。)
 *pInvert 1=反転出力する。(0=反転しない。)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ストロボ信号状態を取得します。

ストロボ信号は、MCM303,GZ4120,GZ320,GZ303 でのみ出力できます。

MCM303 は、 LED1 (= センサ基板側 J2 の Pin2)から出力します。

GZ4120 は、 GPO から出力します。

GZ320,GZ303 は、GPOUT2(= J2 の Pin4)から出力します。

34) MvUsbCam_SetStrobeWidth

DWORD MvUsbCam_SetStrobeWidth(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
dwDat ストロボ幅
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ストロボ幅を設定します。
本関数 CALL 後に取得した入力画像に反映します。

35) MvUsbCam_GetStrobeWidth

DWORD MvUsbCam_GetStrobeWidth(DWORD dwDev, int *pDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat	ストロボ幅 最大値、最小値は、MvUsbCam_GetParamInfo で取得できます。 MvUsbCam_GetParamInfo を CALL する際は、 MVUSBCAM_ParamIndex_STROBE_WIDTH を指定してください。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ストロボ幅を取得します。

36) MvUsbCam_GetStrobeWidthReal

DWORD MvUsbCam_GetStrobeWidthReal(DWORD dwDev, int iDat, double *pDat, int *pUnit)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat	ストロボ幅(Set 値)
*pDat	ストロボ幅(実時間) 最大値、最小値は、MvUsbCam_GetParamInfo で取得できます。 MvUsbCam_GetParamInfo を CALL する際は、 MVUSBCAM_ParamIndex_STROBE_WIDTH を指定してください。
*Unit	ストロボ幅の単位 MVUSBCAM_SHUTTERUNIT_US 等

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ストロボ幅（実時間）を取得します。

37) MvUsbCam_SetStrobeDelayWidth

DWORD MvUsbCam_SetStrobeDelayWidth(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
dwDat ストロボディレイ幅
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

MvUsbCam_GetParamInfo を CALL する際は、
MVUSBCAM_ParamIndex_STROBE_DELAY
を指定してください。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ストロボディレイ幅を設定します。
本関数 CALL 後に取得した入力画像に反映します。

38) MvUsbCam_GetStrobeDelayWidth

DWORD MvUsbCam_GetStrobeDelayWidth(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat ストロボ幅
 最大値、最小値は、MvUsbCam_GetParamInfo で取得できます。
 MvUsbCam_GetParamInfo を CALL する際は、
 MVUSBCAM_ParamIndex_STROBE_DELAY
 を指定してください。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ストロボディレイ幅を取得します。

39) MvUsbCam_GetStrobeDelayWidthReal

DWORD MvUsbCam_GetStrobeDelayWidthReal(DWORD dwDev,
int iDat, double *pDat, int *pUnit)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat	ストロボ幅(Set 値)
*pDat	ストロボ幅(実時間) 最大値、最小値は、MvUsbCam_GetParamInfo で取得できます。 MvUsbCam_GetParamInfo を CALL する際は、 MVUSBCAM_ParamIndex_STROBE_DELAY を指定してください。
*Unit	ストロボ幅の単位 MVUSBCAM_SHUTTERUNIT_US 等

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ストロボディレイ幅（実時間）を取得します。

40) MvUsbCam_SetSensorOffset

DWORD MvUsbCam_SetSensorOffset(DWORD dwDev, int iX, int iY)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iX	センサ出力 X オフセット
iY	センサ出力 Y オフセット

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

センサオフセット値の設定をします。

41) MvUsbCam_GetSensorOffset

DWORD MvUsbCam_GetSensorOffset(DWORD dwDev, int *pX, int *pY)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pX	センサ出力 X オフセット
*pY	センサ出力 Y オフセット

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

センサオフセット値を取得します。

42) MvUsbCam_SetWhiteBalanceAuto

DWORD MvUsbCam_SetWhiteBalanceAuto(DWORD dwDev, BOOL bEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bDat 自動ホワイトバランス値(TRUE : 自動 FALSE : 非自動)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

自動ホワイトバランス値を設定します。
本関数CALL後に取得した入力画像に反映します。

43) MvUsbCam_GetWhiteBalanceAuto

DWORD MvUsbCam_GetWhiteBalanceAuto(DWORD dwDev, BOOL *pEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat 自動ホワイトバランス値(TRUE : 自動、FALSE : 非自動)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

自動ホワイトバランス値を取得します。

44) MvUsbCam_SetWhiteBalanceRed

DWORD MvUsbCam_SetWhiteBalanceRed(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトバランス 赤レベル
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ホワイトバランス 赤レベルを設定します。
本関数 CALL 後に取得した入力画像に反映します。

45) MvUsbCam_GetWhiteBalanceRed

DWORD MvUsbCam_GetWhiteBalanceRed(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトバランス 赤レベル
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ホワイトバランス 赤レベルを取得します。

46) MvUsbCam_SetWhiteBalanceGreen

DWORD MvUsbCam_SetWhiteBalanceGreen(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイ トバ ラ ンス 緑 レベ ル
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ホワイ トバ ラ ンス 緑 レベ ルを 設 定 し ます。
本関数 CALL 後に取得した入力画像に反映します。

47) MvUsbCam_GetWhiteBalanceGreen

DWORD MvUsbCam_GetWhiteBalanceGreen(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ~ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

iDat ホワイ トバランス 緑レベル

最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ホワイ トバランス 緑レベルを取得します。

48) MvUsbCam_SetWhiteBalanceBlue

DWORD MvUsbCam_SetWhiteBalanceBlue(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトバランス 青レベル
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ホワイトバランス 青レベルを設定します。
本関数 CALL 後に取得した入力画像に反映します。

49) MvUsbCam_GetWhiteBalanceBlue

DWORD MvUsbCam_GetWhiteBalanceBlue(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトバランス 青レベル
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ホワイトバランス 青レベルを取得します。

50) MvUsbCam_SetWhiteBalanceRGB

DWORD MvUsbCam_SetWhiteBalanceRGB(DWORD dwDev, int iR, int iG, int iB)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iR	ホワイトバランス 赤レベル
iG	ホワイトバランス 緑レベル
iB	ホワイトバランス 青レベル

最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ホワイトバランス 赤緑青レベルを一括で設定します。
本関数 CALL 後に取得した入力画像に反映します。

51) MvUsbCam_GetWhiteBalanceRGB

DWORD MvUsbCam_GetWhiteBalanceRGB(DWORD dwDev, int *pR, int *pG, int *pB)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pR	ホワイトバランス 赤レベル
*pG	ホワイトバランス 緑レベル
*pB	ホワイトバランス 青レベル

最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ホワイトバランス 赤緑青レベルを一括で取得します。

52) MvUsbCam_WhiteBalanceUpdate

DWORD MvUsbCam_WhiteBalanceUpdate(DWORD dwDev)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ホワイトバランス更新をします。

本関数は、MCM320／GZ320 専用関数です。

53) MvUsbCam_SetWhitePointX

DWORD MvUsbCam_SetWhitePointX(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトポイント X 値
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

sRGB 空間上のホワイトポイント X 値を設定します。

「ホワイトポイント X 値」と「ホワイトポイント X オフセット値」の合計値が、
ホワイトポイント X 調整値として、ホワイトバランス画質に反映します。
本関数 CALL 後に取得した入力画像に反映します。

ただし、以下の 2 つの条件が必要です。

自動ホワイトバランスが無効になっている事。
色温度設定が有効になっている事。

本関数は、GZ320 専用関数です。

54) MvUsbCam_GetWhitePointX

DWORD MvUsbCam_GetWhitePointX(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトポイント X 値を返します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

sRGB 空間上のホワイトポイント X 値を取得します。

本関数は、以前に、MvUsbCam_SetWhitePointX で設定した値を取得します。

本関数は、GZ320 専用関数です。

55) MvUsbCam_GetWhitePointX_now

DWORD MvUsbCam_GetWhitePointX_now(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat ホワイトポイント X 現在値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

sRGB 空間上のホワイトポイント X 現在値を取得します。

MvUsbCam_WhiteBalanceUpdate 実行の前後で、値が変化する事があります。

本関数は、GZ320 専用関数です。

56) MvUsbCam_SetWhitePointY

DWORD MvUsbCam_SetWhitePointY(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトポイントY 値
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

sRGB 空間上のホワイトポイントY 値を設定します。

「ホワイトポイントY 値」と「ホワイトポイントY オフセット値」の合計値が、
ホワイトポイントY 調整値として、ホワイトバランス画質に反映します。
本関数 CALL 後に取得した入力画像に反映します。

ただし、以下の 2 つの条件が必要です。

自動ホワイトバランスが無効になっている事。
色温度設定が有効になっている事。

本関数は、GZ320 専用関数です。

57) MvUsbCam_GetWhitePointY

DWORD MvUsbCam_GetWhitePointY(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトポイント Y 値を返します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

sRGB 空間上のホワイトポイント Y 値を取得します。

以前に、MvUsbCam_SetWhitePointY で設定した値を取得します。

本関数は、GZ320 専用関数です。

58) MvUsbCam_GetWhitePointY_now

DWORD MvUsbCam_GetWhitePointY_now(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat ホワイトポイント Y 現在値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

sRGB 空間上のホワイトポイント Y 現在値を取得します。

MvUsbCam_WhiteBalanceUpdate 実行の前後で、値が変化する事があります。

本関数は、GZ320 専用関数です。

59) MvUsbCam_SetWhitePointXOffset

DWORD MvUsbCam_SetWhitePointXOffset(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトポイントXオフセット値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

sRGB 空間上のホワイトポイントXオフセット値を設定します。

「ホワイトポイントX値」と「ホワイトポイントXオフセット値」の合計値が、
ホワイトポイントX調整値として、ホワイトバランス画質に反映します。
本関数 CALL 後に取得した入力画像に反映します。

ただし、以下の2つの条件が必要です。

自動ホワイトバランスが無効になっている事。
色温度設定が有効になっている事。

本関数は、GZ320 専用関数です。

60) MvUsbCam_GetWhitePointXOffset

DWORD MvUsbCam_GetWhitePointXOffset(DWORD dwDev, int *pDat)

引数

dwDev	デバイス番号(0 ~ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat	ホワイトポイント X オフセット値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

sRGB 空間上のホワイトポイント X オフセット値を取得します。

以前、MvUsbCam_SetWhitePointXOffset で設定した値を返します。

本関数は、GZ320 専用関数です。

61) MvUsbCam_GetWhitePointXOffset_now

DWORD MvUsbCam_GetWhitePointXOffset_now(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat ホワイトポイント X オフセット 現在値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

sRGB 空間上のホワイトポイント X オフセット 現在値を取得します。

MvUsbCam_WhiteBalanceUpdate 実行の前後で、値が変化する事があります。

本関数は、GZ320 専用関数です。

62) MvUsbCam_SetWhitePointYOffset

DWORD MvUsbCam_SetWhitePointYOffset(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat ホワイトポイントYオフセット値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

sRGB 空間上のホワイトポイントYオフセット値を設定します。

「ホワイトポイントY値」と「ホワイトポイントYオフセット値」の合計値が、
ホワイトポイントY調整値として、ホワイトバランス画質に反映します。
本関数 CALL 後に取得した入力画像に反映します。

ただし、以下の2つの条件が必要です。

自動ホワイトバランスが無効になっている事。
色温度設定が有効になっている事。

本関数は、GZ320 専用関数です。

63) MvUsbCam_GetWhitePointYOffset

DWORD MvUsbCam_GetWhitePointYOffset(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat ホワイトポイント Y オフセット値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

sRGB 空間上のホワイトポイント Y オフセット値を取得します。

以前、MvUsbCam_SetWhitePointYOffset で設定した値を返します。

本関数は、GZ320 専用関数です。

64) MvUsbCam_GetWhitePointYOffset_now

DWORD MvUsbCam_GetWhitePointYOffset_now(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat ホワイトポイントYオフセット値現在値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

sRGB 空間上のホワイトポイントYオフセット現在値を取得します。

MvUsbCam_WhiteBalanceUpdate 実行の前後で、値が変化する事があります。

本関数は、GZ320 専用関数です。

65) MvUsbCam_SetColorTemperatureEnable

DWORD MvUsbCam_SetColorTemperatureEnable(DWORD dwDev, BOOL bEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bDat 色温度の有効無効値(TRUE : 色温度有効 FALSE : 色温度無効)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

色温度の有効無効値を設定します。
本関数CALL後に取得した入力画像に反映します。
本関数は、GZ4120専用関数です。

66) MvUsbCam_GetColorTemperatureEnable

DWORD MvUsbCam_GetColorTemperatureEnable (DWORD dwDev, BOOL *pEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat 色温度の有効無効値(TRUE : 色温度有効 FALSE : 色温度無効)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

色温度の有効無効値を取得します。
 本関数は、GZ4120 専用関数です。

67) MvUsbCam_SetColorTemperatureValue

DWORD MvUsbCam_SetColorTemperatureValue (DWORD dwDev, DWORD dwTemperature)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
dwTemperature	色温度値
	dwTemperature = 4000 ... 4000K
	dwTemperature = 6000 ... 6000K

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

色温度値を設定します。

本関数CALL後に取得した入力画像に反映します。

本関数は、GZ4120専用関数です

68) MvUsbCam_GetColorTemperatureValue

DWORD MvUsbCam_GetColorTemperatureValue (DWORD dwDev, DWORD *pTemperature)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

* pTemperature 色温度値

dwTemperature = 4000 ... 4000K

dwTemperature = 6000 ... 6000K

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

色温度値を取得します。

本関数は、GZ4120 専用関数です。

69) MvUsbCam_SetWBColorConfig_each

DWORD MvUsbCam_SetWBColorConfig_each (DWORD dwDev,
int Area, WORD wGain, WORD wHue)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
Area	CbCr 空間上のエリア番号(1=第 1 エリア、 2=第 2 エリア、... 6=第 6 エリア)
wGain	ゲイン値 ([15:10] 未使用、[9:0] Q2 フォーマット)
wHue	ヒュー値 ([15:13] 未使用、[12:0] Q4 フォーマット)

wGain = 0x000 の時、 最大ゲイン値=0.00 を指します。

wGain = 0x100 の時、 デフォルトゲイン値=1.00 を指します。

wGain = 0x200 の時、 最大ゲイン値=2.00 を指します。

wHue=0x14C の時、 最大ヒュー値 = 180° を示します。

wHue=0x100 の時、 デフォルトヒュー値 = 0° を示します。

wHue=0x200 の時、 最小ヒュー値 = -180° を示します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

カラーバランス調整値を（エリア指定で）設定します。

本関数は、CALL直後に、入力画像に反映します。

本関数は、GZ4120専用関数です。

70) MvUsbCam_GetWBColorConfig_each

DWORD MvUsbCam_GetWBColorConfig_each (DWORD dwDev,
int Area, WORD *pGain ,WORD *pHue)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
Area	CbCr 空間上のエリア値(1=第 1 エリア、 2=第 2 エリア、... 6=第 6 エリア)
*pGain	ゲイン値 ([15:10] 未使用、[9:0] Q2 フォーマット)
*pHue	ヒュー値 ([15:13] 未使用、[12:0] Q4 フォーマット)

wGain = 0x000 は、	最大ゲイン値=0.00 を示します。
wGain = 0x100 は、	デフォルトゲイン値=1.00 を示します。
wGain = 0x200 は、	最大ゲイン値=2.00 を示します。

wHue=0x14C は、	最大ヒュー値 = 180° を示します。
wHue=0x100 は、	デフォルトヒュー値 = 0° を示します。
wHue=0x200 は、	最小ヒュー値 = -180° を示します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

カラーバランス調整値を (エリア指定で) 取得します。

本関数は、GZ4120 専用関数です。

71) MvUsbCam_SetWBColorConfig_all

DWORD MvUsbCam_SetWBColorConfig_all (DWORD dwDev,
MVUSBCAM_WB_ColorGain_PARA *pDatIn)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

*pDatIn MVUSBCAM_WB_ColorGain_PARA 構造体
CbCr 空間上の全エリアのゲイン値、ヒュー値を含みます。
(詳しくは、取扱説明書 for GZ4120 を参照の事。)

pDatIn->wAreaNGain = 0x000 は、 最大ゲイン値=0.00 を示します。
pDatIn->wAreaNGain = 0x100 は、 デフォルトゲイン値=1.00 を示します。
pDatIn->wAreaNGain = 0x200 は、 最大ゲイン値=2.00 を示します。

pDatIn->wAreaNHue=0x14C は、 最大ヒュー値 = 180° を示します。
pDatIn->wAreaNHue=0x100 は、 デフォルトヒュー値 = 0° を示します。
pDatIn->wAreaNHue=0x200 は、 最小ヒュー値 = -180° を示します。

ただし、pDatIn->wAreaNGain の N は、1 ～ 6 を表します。
また、pDatIn->wAreaNHue の N は、1 ～ 6 を表します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

全エリアのカラーバランス調整値を設定します。
本関数は、CALL直後に、入力画像に反映します。
本関数は、GZ4120専用関数です。

72) MvUsbCam_GetWBColorConfig_all

DWORD MvUsbCam_GetWBColorConfig_all (DWORD dwDev,
MVUSBCAM_WB_ColorGain_PARA *pDatIn)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

*pDatIn MVUSBCAM_WB_ColorGain_PARA 構造体
CbCr 空間上の全エリアのゲイン値、ヒュー値を含みます。
(詳しくは、取扱説明書 for GZ4120 を参照の事。)

pDatIn->wAreaNGain = 0x000 は、 最大ゲイン値=0.00 を示します。
pDatIn->wAreaNGain = 0x100 は、 デフォルトゲイン値=1.00 を示します。
pDatIn->wAreaNGain = 0x200 は、 最大ゲイン値=2.00 を示します。

pDatIn->wAreaNHue=0x14C は、 最大ヒュー値 = 180° を示します。
pDatIn->wAreaNHue=0x100 は、 デフォルトヒュー値 = 0° を示します。
pDatIn->wAreaNHue=0x200 は、 最小ヒュー値 = -180° を示します。

ただし、pDatIn->wAreaNGain の N は、1 ～ 6 を表します。
また、pDatIn->wAreaNHue の N は、1 ～ 6 を表します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

全エリアのカラーバランス調整値を取得します。
本関数は、GZ4120 専用関数です。

73) MvUsbCam_SetExposureAuto

DWORD MvUsbCam_SetExposureAuto(DWORD dwDev, BOOL bDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bDat 自動露光設定値(TRUE : 有効 FALSE : 無効)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

自動露光を設定します。
本関数 CALL 後に取得した入力画像に反映します。

74) MvUsbCam_GetExposureAuto

DWORD MvUsbCam_GetExposureAuto(DWORD dwDev, BOOL *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat 自動露光設定値(TRUE : 有効 FALSE : 無効)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

自動露光状態を取得します。

75) MvUsbCam_SetAeTarget

DWORD MvUsbCam_SetAeTarget(DWORD dwDev, int iAeTarget)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iAeTarget 自動露光ターゲット値

値が小さい時は、暗くなります。
値が大きい時は、明るくなります。
ただし、自動露光時のみ有効です。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

自動露光ターゲット値を設定します。
本関数 CALL 後に取得した入力画像に反映します。
本関数は、GZ4120 専用関数です。

76) MvUsbCam_GetAeTarget

DWORD MvUsbCam_GetAeTarget(DWORD dwDev, int *pAeTarget)

引数

dwDev デバイス番号(0 ~ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

*pAeTarget 自動露光ターゲット値

値が小さい時は、暗くなります。
 値が大きい時は、明るくなります。
 ただし、自動露光時のみ有効です。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

自動露光ターゲット値を取得します。

本関数は、GZ4120 専用関数です。

77) MvUsbCam_SetFrameRateFix

DWORD MvUsbCam_SetFrameRateFix(DWORD dwDev, BOOL bDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bDat フレームレート固定設定値(TRUE : 有効 FALSE : 無効)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

フレームレート固定値を設定します。
自動露光状態の時のみ、有効となります。
本関数 CALL 後に取得した入力画像に反映します。

78) MvUsbCam_GetFrameRateFix

DWORD MvUsbCam_GetFrameRateFix(DWORD dwDev, BOOL *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bDat フレームレート固定設定値(TRUE : 固定 FALSE : 非固定)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

フレームレート固定状態を取得します。

79) MvUsbCam_SetAnalogGainAuto

DWORD MvUsbCam_SetAnalogGainAuto (DWORD dwDev, BOOL bDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bDat アナログゲインオート値(TRUE : オート FALSE : 非オート)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

アナログゲインオート値を設定します。

80) MvUsbCam_GetAnalogGainAuto

DWORD MvUsbCam_GetAnalogGainAuto (DWORD dwDev, BOOL *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bDat アナログゲインオート値(TRUE : オート FALSE : 非オート)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

アナログゲインオート値を取得します。

81) MvUsbCam_SetShutterSpeed

DWORD MvUsbCam_SetShutterSpeed(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
dwDat シャッタースピード
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

シャッタースピードを設定します。
本関数 CALL 後に取得した入力画像に反映します。

82) MvUsbCam_GetShutterSpeed

DWORD MvUsbCam_GetShutterSpeed(DWORD dwDev, int *pDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat	シャッタースピード 最大値、最小値は、MvUsbCam_GetParamInfo で取得できます。 MvUsbCam_GetParamInfo を CALL する際は、 MVUSBCAM_ParamIndex_EC_SHUTTER を指定してください。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

シャッタースピードを取得します。

83) MvUsbCam_GetShutterSpeedReal

DWORD MvUsbCam_GetShutterSpeedReal(DWORD dwDev, int iDat, double *pDat, int *pUnit)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat	シャッタースピード (Set 値)
*pDat	シャッタースピード (実時間) 最大値、最小値は、MvUsbCam_GetParamInfo で取得できます。 MvUsbCam_GetParamInfo を CALL する際は、 MVUSBCAM_ParamIndex_EC_SHUTTER を指定してください。
*Unit	シャッタースピードの単位 MVUSBCAM_SHUTTERUNIT_US 等

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

シャッタースピード (実時間) を取得します。

84) MvUsbCam_SetDigitalGain

DWORD MvUsbCam_SetDigitalGain(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
dwDat ゲイン設定値
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

デジタルゲイン値を設定します。
本関数 CALL 後に取得した入力画像に反映します。

85) MvUsbCam_GetDigitalGain

DWORD MvUsbCam_GetDigitalGain(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat ゲイン設定値
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

デジタルゲイン値を取得します。

86) MvUsbCam_SetAnalogGain

DWORD MvUsbCam_SetAnalogGain(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat アナログゲイン設定値
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

アナログゲイン値を設定します。
本関数 CALL 後に取得した入力画像に反映します。

87) MvUsbCam_GetAnalogGain

DWORD MvUsbCam_GetAnalogGain(DWORD dwDev, int *pDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat	アナログゲイン設定値
iDat	アナログゲイン設定値

最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

アナログゲイン値を取得します。

88) MvUsbCam_CameraBusyCheck

DWORD MvUsbCam_CameraBusyCheck(DWORD dwDev, BOOL *pBusy)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pBusy カメラビジー状態 (1 = ビジー状態、0 = アイドル状態)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

カメラビジー状態を取得します。
 MCM320 専用関数です。

89) MvUsbCam_CameraBusyWait

DWORD MvUsbCam_CameraBusyWait(DWORD dwDev, BOOL *pBusy, int iTimeout)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pBusy	カメラビジー状態 (1 = ビジー状態、0 = アイドル状態)
iTimeout	オーバータイム時間(msec)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

カメラビジー状態が終わるのを待ちます。

iTimeout(msec)時間たったら、必ず終了します。

本関数の制御を止めたい場合は、別スレッドから、

MvUsbCam_CameraBusyWaitCancel を C A L L してください。

MCM320 専用関数です。

90) MvUsbCam_CameraBusyWaitCancel

DWORD MvUsbCam_CameraBusyWaitCancel(DWORD dwDev)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

MvUsbCam_CameraBusyWait を途中終了させる時に C A L L します。

MvUsbCam_CameraBusyWait と違うスレッドから、C A L L する必要があります。

MCM320 専用関数です。

91) MvUsbCam_SetShutterMaxMin

DWORD MvUsbCam_SetShutterMaxMin(DWORD dwDev, WORD wMax, WORD wMin)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
wMax	シャッター上限値
wMin	シャッター下限値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

シャッタースピードのソフトウェアリミット（上限値／下限値）を設定します。
 本関数は、自動露光時や、シャッタースピード設定時の処理に影響があります。
 また、自動露光時に限り、本関数 CALL 直後の入力画像に反映します。
 本関数は、GZ4120 専用関数です。

92) MvUsbCam_GetShutterMaxMin

DWORD MvUsbCam_GetShutterMaxMin(DWORD dwDev, WORD *pMax, WORD *pMin)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pMax	シャッター上限値
*pMin	シャッター下限値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

シャッタースピードのソフトウェアリミット（上限値／下限値）を取得します。

本関数は、GZ4120 専用関数です

93) MvUsbCam_SetAnaGainMaxMin

DWORD MvUsbCam_SetAnaGainMaxMin(DWORD dwDev, WORD wMax, WORD wMin)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
wMax	アナログゲイン上限値
wMin	アナログゲイン下限値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

アナログゲインのソフトウェアリミット（上限値／下限値）を設定します。
 本関数は、自動露光時や、アナログゲインスピード設定時の処理に影響があります。
 また、自動露光時に限り、本関数 CALL 直後の入力画像に反映します。
 本関数は、GZ4120 専用関数です。

94) MvUsbCam_GetAnaGainMaxMin

DWORD MvUsbCam_GetAnaGainMaxMin(DWORD dwDev, WORD *pMax, WORD *pMin)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pMax	アナログゲイン上限値
*pMin	アナログゲイン下限値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

アナログゲインのソフトウェアリミット（上限値／下限値）を取得します。

本関数は、GZ4120 専用関数です

95) MvUsbCam_SetDigGainMaxMin

DWORD MvUsbCam_SetDigGainMaxMin(DWORD dwDev, WORD wMax, WORD wMin)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
wMax	デジタルゲイン上限値
wMin	デジタルゲイン下限値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

デジタルゲインのソフトウェアリミット（上限値／下限値）を設定します。
 本関数は、自動露光時や、デジタルゲインスピード設定時の処理に影響があります。
 また、自動露光時に限り、本関数 CALL 直後の入力画像に反映します。
 本関数は、GZ4120 専用関数です。

96) MvUsbCam_GetDigGainMaxMin

DWORD MvUsbCam_GetDigGainMaxMin(DWORD dwDev, WORD *pMax, WORD *pMin)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pMax	デジタルゲイン上限値
*pMin	デジタルゲイン下限値

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

デジタルゲインのソフトウェアリミット（上限値／下限値）を取得します。

本関数は、GZ4120 専用関数です

97) MvUsbCam_SetFrameRateCtrl

DWORD MvUsbCam_SetFrameRateCtrl(DWORD dwDev, int iMode, int iDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iMode	画像サイズ設定値。以下の設定値が有効です。 MCM320_FrameRateCtrl_A MCM320_FrameRateCtrl_C
iDat	フレームレート設定値

MvUsbCam_GetFrameRateInfo により、設定値を取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

最大フレームレート値を設定します。

実際のフレームレートは、自動露光状態、シャッタースピード、撮像対象の明るさ等の影響により変化する事があります。

98) MvUsbCam_GetFrameRateCtrl

DWORD MvUsbCam_GetFrameRateCtrl(DWORD dwDev, int iMode, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iMode 画像サイズ設定値。以下の設定値が有効です。

MVUSBCAM_FrameRateCtrl_A
MVUSBCAM_FrameRateCtrl_B
MVUSBCAM_FrameRateCtrl_C

*pDat フレームレート設定値

MvUsbCam_GetFrameRateInfo により、設定値を取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

最大フレームレート値を取得します。

実際のフレームレートは、自動露光状態、シャッタースピード、撮像対象の明るさ等の影響により変化する事があります。

99) MvUsbCam_SetFrameRateCommon

DWORD MvUsbCam_SetFrameRateCommon(DWORD dwDev, BOOL bEnable, double dFps)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bEnable	1 = 共通フレームレート有効、0 = 無効
dFps	共通フレームレート値 (単位は、1/1000 Fps)
	dFps = 10000 は、10fps を示します。
	dFps = 25500 は、25.5fps を示します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

共通フレームレートを設定します。

共通フレームレート有効時は、GazooCapture や AMCap 等で指定するフレームレートを無視して、フレームレート値 dFps で動作します。

ただし、現サイズ設定で、出力可能なフレームレートでない場合は、画が出ない事があります。

また、実際のフレームレートは、自動露光状態、シャッタースピード、撮像対象の明るさ等の影響により変化する事があります。

本関数 CALL 後に取得した入力画像に反映します。

本関数は、GZ4120 専用関数です。

100) MvUsbCam_GetFrameRateCommon

DWORD MvUsbCam_GetFrameRateCommon(DWORD dwDev, BOOL *pEnable, double *pFps)

引数

dwDev	デバイス番号(0 ~ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pEnable	1 = 共通フレームレート有効、0 = 無効
*pFps	共通フレームレート値 (単位は、1/1000 fps) dFps = 10000 は、10fps を示します。 dFps = 25500 は、25.5fps を示します。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

共通フレームレート設定値を取得します。

共通フレームレート有効時は、GazooCapture や AMCap の指定フレームレートを無視して、フレームレート値 dFps で動作します。

ただし、現サイズ設定で、出力可能フレームレートでない場合は、画が出ない事があります。

また、実際のフレームレートは、自動露光状態、シャッタースピード、撮像対象の明るさ等の影響により変化する事があります。

本関数は、GZ4120 専用関数です。

101) MvUsbCam_DefaultGains

DWORD MvUsbCam_DefaultGains(DWORD dwDev)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

画質調整値を初期値に戻します。

ブライトネス、コントラスト等のすべての画質調整値を、初期値に戻します。

本関数 CALL 後に取得した入力画像に反映します。

102) MvUsbCam_SaveGains

DWORD MvUsbCam_SaveGains(DWORD dwDev)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

カメラに、画質調整値群をセーブします。

103) MvUsbCam_SensorInit

DWORD MvUsbCam_SensorInit(DWORD dwDev, BOOL bPwr)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bPwr カメラセンサ部の電源制御を指定(1 = 電源 OFF あり、0=電源 ON なし)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

カメラの初期化（電源OFFを含むレジスタ初期化）をします。
MCM320 専用関数です。

104) MvUsbCam_GetSensorInitCount

DWORD MvUsbCam_GetSensorInitCount(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat MvUsbCam_SensorInit の C A L L 回数

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

MvUsbCam_SensorInit の C A L L 回数を取得します。

105) MvUsbCam_SetGpio

DWORD MvUsbCam_SetGpio(DWORD dwDev, int iNo, int iDat)

引数

dwDev	デバイス番号(0 ~ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iNo	GPIO 番号 MVUSBCAM_GPIO1, MVUSBCAM_GPIO_TrgIn 等
iDat	GPIO 出力値(1=High, 0=Low)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

GPIO の出力をします。

MCM-303 の場合は、以下の動作が可能です。

GPIO1,GPIO2,GPIO3, GPIO4, GPIO5, GPIO6 の入出力。

USR_TRIG_IN(USB 基板側 USB_J3)の入力。

MCM-320 の場合は、以下の動作が可能です。

GPIO1,GPIO2,GPIO3 の入出力。

USR_TRIG_IN(USB 基板側 J3)の入力。

106) MvUsbCam_GetGpio

DWORD MvUsbCam_GetGpio(DWORD dwDev, int iNo, int *pDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iNo	GPIO 番号 MVUSBCAM_GPIO1, MVUSBCAM_GPIO_TrgIn 等
*pDat	GPIO 状態値の取得(1=High, 0=Low)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

GPIO 状態値の取得をします。

MCM-303 の場合は、以下の動作が可能です。

GPIO1,GPIO2,GPIO3, GPIO4, GPIO5, GPIO6 の入出力。

USR_TRIG_IN(USB 基板側 USB_J3)の入力。

MCM-320 の場合は、以下の動作が可能です。

GPIO1,GPIO2,GPIO3 の入出力。

USR_TRIG_IN(USB 基板側 J3)の入力。

107) MvUsbCam_SetGpioAll

DWORD MvUsbCam_SetGpioAll(DWORD dwDev, MVUSBCAM_SET_GPIO_ALL_INFO *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat MVUSBCAM_SET_GPIO_ALL_INFO 構造体

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

GPIO の一括出力をします。

MVUSBCAM_SET_GPIO_ALL_INFO 構造体要素の dwVersion を、
 MVUSBCAM_SET_GPIO_ALL_INFO_VERSION に変更してから、CALL してください。

108) MvUsbCam_GetGpioAll

DWORD MvUsbCam_GetGpioAll(DWORD dwDev, MVUSBCAM_SET_GPIO_ALL_INFO *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat MVUSBCAM_SET_GPIO_ALL_INFO 構造体

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

GPIO 値を一括に取得します。

MVUSBCAM_SET_GPIO_ALL_INFO 構造体要素の dwVersion を、
 MVUSBCAM_SET_GPIO_ALL_INFO_VERSION に変更してから、CALL してください。

109) MvUsbCam_SetGpioSetting

DWORD MvUsbCam_SetGpioSetting(DWORD dwDev, int iNo, BOOL bOutput, BOOL bInitHigh)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bOutput	入出力方向の設定(1=出力, 0=入力)
bInitHigh	起動時の出力状態(1=High, 0=Low)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

GPIO の設定をします。

GPIO の入出力を切り替えには、以下の 3 つの作業が必要です。

Step1. 関数 MvUsbCam_SetGpioSetting の CALL。
(または、GainSetup の GPIO メニューで同等の操作。)

Step2. 関数 MvUsbCam_SaveGains の CALL。
(または、GainSetup で、Save クリック。)

Step3. USB 接続の抜き差し。

110) MvUsbCam_GetGpioSetting

DWORD MvUsbCam_GetSetting(DWORD dwDev, int iNo, BOOL *pOutput, BOOL *pInitHigh)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pOutput 入出力方向の設定(1=出力, 0=入力)
 *pInitHigh 起動時の出力状態(1=High, 0=Low)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

GPIO 設定値を取得します。

111) MvUsbCam_SetTriggerSetting

DWORD MvUsbCam_SetTriggerSetting(DWORD dwDev, int iDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat 外部トリガー設定値
 MCM303,GZ320,GZ303 の場合は、(1=トリガー待ち、0=待たない)
 GZ3970 の場合は、(2=SW トリガ待ち、1=HW トリガー待ち、0=待たない)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

外部トリガーを設定します。

MCM-303, GZ3970,GZ320,GZ303 専用関数です。

112) MvUsbCam_GetTriggerSetting

DWORD MvUsbCam_GetTriggerSetting(DWORD dwDev, int *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat 外部トリガー設定値
 MCM303,GZ320,GZ303 の場合は、(1=トリガー待ち、0=待たない)
 GZ3970 の場合は、(2=SW トリガ待ち、1=HW トリガー待ち、0=待たない)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

外部トリガー値を取得します。

MCM-303, GZ3970,GZ320,GZ303 専用関数です。

113) MvUsbCam_SoftTriggerON

DWORD MvUsbCam_SoftTriggerON(DWORD dwDev)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ソフトウェアトリガーを ON にします。

本館数の CALL タイミングで画像が取り込まれます。

GZ3970 専用関数です。

使用例は、以下の通りです。

MvUsbCam_SetTriggerSetting(dwDev, 0) ... トリガーなしに設定。

GazooCapture で「取り込み」実行。 ... カメラ画像表示が更新続ける。

MvUsbCam_SetTriggerSetting(dwDev, 2) ... ソフトウェアトリガー待ち。

GazooCapture で、カメラ画像表示更新が止まる。

MvUsbCam_SoftTriggerON(dwDev) ... ソフトウェアトリガー実行。

GazooCapture で、カメラ画像表示が更新され、すぐ止まる。

114) MvUsbCam_GetGpioInfo

DWORD MvUsbCam_GetGpioInfo(DWORD dwDev, MVUSBCAM_GPIO_INFO *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat GPIO (有効無効) 情報構造体

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

GPIO (有効無効) 情報を取得します。

MVUSBCAM_GPIO_INFO 構造体要素の dwVersion を、
 MVUSBCAM_GPIO_INFO_VERSION に変更してから、CALL してください。

115) MvUsbCam_SetGpioPath

DWORD MvUsbCam_SetGpioPath(DWORD dwDev, int iMode, BOOL bEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

iMode GPIO PATH モード
 0 : PWM
 1 : Flash/Shutter

bEnb GPIO PATH 有効無効指定

iMode = 0, かつ bEnb = 1 ならば、 PWM 有効, GPOUT3/4 無効

iMode = 0, かつ bEnb = 0 ならば、 PWM 無効, GPOUT3/4 有効

iMode = 1, かつ bEnb = 1 ならば、 SHUTTER/STROBE 有効, GPOUT1/2 無効

iMode = 1, かつ bEnb = 0 ならば、 SHUTTER/STROBE 無効, GPOUT1/2 有効

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

GPIO PATH 設定をします。

本関数は、GZ320／GZ303 専用関数です。

GZ320／GZ303 では、本関数を使用する事で、GPOUT(GPOUT1 ～ GPOUT4)として動作させるか、SHUTTER/STROBE や、PWM として動作させるかを選択する事ができます。

iMode = 0 の時

bEnb	PWM	GPOUT3	GPOUT4
0	無効	有効	有効
1	有効	無効	無効

iMode = 1 の時

bEnb	SHUTTER	STROBE	GPOUT1	GPOUT2
0	無効	無効	有効	有効
1	有効	有効	無効	無効

116) MvUsbCam_GetGpioPath

DWORD MvUsbCam_GetGpioPath(DWORD dwDev, int iMode, BOOL *bEnb)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

iMode GPIO PATH モード
 0 : PWM
 1 : Flash/Shutter

*pEnb GPIO PATH 有効無効指定

iMode = 0, かつ bEnb = 1 ならば、 PWM 有効, GPOUT3/4 無効

iMode = 0, かつ bEnb = 0 ならば、 PWM 無効, GPOUT3/4 有効

iMode = 1, かつ bEnb = 1 ならば、 SHUTTER/STROBE 有効, GPOUT1/2 無効

iMode = 1, かつ bEnb = 0 ならば、 SHUTTER/STROBE 無効, GPOUT1/2 有効

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います

MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

GPIO PATH 設定値を取得します。

本関数は、GZ320／GZ303 専用関数です。

117) MvUsbCam_WriteUserData

DWORD MvUsbCam_WriteUserData(DWORD dwDev, MVUSBCAM_UserData_DAT *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat ユーザーデータアクセス用構造体

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ユーザーデータの書き込みをします。
ユーザーデータ 3 2 バイトを、一括で、書き込みます。

MVUSBCAM_UserData_DAT 構造体要素の dwVersion を、
MVUSBCAM_UserData_DAT_VERSION に変更してから、CALL してください。

118) MvUsbCam_ReadUserData

DWORD MvUsbCam_ReadUserData(DWORD dwDev, MVUSBCAM_UserData_DAT *pDat)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
 *pDat ユーザーデータアクセス用構造体

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
 MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
 MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ユーザーデータの読み出しをします。
 ユーザーデータ 3 2 バイトを、一括で、読み出します。

MVUSBCAM_UserData_DAT 構造体要素の dwVersion を、
 MVUSBCAM_UserData_DAT_VERSION に変更してから、CALL してください。

119) MvUsbCam_StartUserTrgMonitor

DWORD MvUsbCam_StartUserTrgMonitor(DWORD dwDev)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーのモニタを開始します。

ユーザートリガーモニター状態(* 1)で、トリガー検知すると、
MvUsbCam_SetUserTrgChangeCB で指定した関数にコールバックされます。

*** 1** ... MvUsbCam_StartUserTrgMonitor が C A L L されてから、
MvUsbCam_StopUserTrgMonitor が C A L L されるまでが、
ユーザートリガーモニター状態とします。

120) MvUsbCam_StopUserTrgMonitor

DWORD MvUsbCam_StopUserTrgMonitor(DWORD dwDev)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーのモニタを停止します。

ユーザートリガーモニター状態(* 1)で、トリガー検知すると、
MvUsbCam_SetUserTrgChangeCB で指定した関数にコールバックされます。

*** 1** ... MvUsbCam_StartUserTrgMonitor が C A L L されてから、
MvUsbCam_StopUserTrgMonitor が C A L L されるまでが、
ユーザートリガーモニター状態とします。

121) MvUsbCam_GetUserTrgData

DWORD MvUsbCam_GetUserTrgData(DWORD dwDev, DWORD *pTrgHitNum, BOOL *pLevel)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pTrgHitNum	トリガー回数
*pLevel	トリガー受信時の I/O レベル(0=LOW, 1=HIGH)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーのデータを取得します。

122) MvUsbCam_ClearUserTrgData

DWORD MvUsbCam_ClearUserTrgData(DWORD dwDev)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーのデータをクリアします。

123) MvUsbCam_SetUserTrgEdge

DWORD MvUsbCam_SetUserTrgEdge(DWORD dwDev, BOOL bRising)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
bRising ユーザートリガー条件 (0=立ち下がり、 1 =立ち上がり)

ユーザートリガー条件に従って、ユーザートリガー情報がコールバック通知されます。
(ただし、ユーザートリガーモード= 0 ないし 2 の時のみに限ります。)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

ユーザートリガー条件を設定します。

本関数CALL直後から、ユーザートリガー動作が更新されます。

124) MvUsbCam_GetUserTrgEdge

DWORD MvUsbCam_GetUserTrgEdge(DWORD dwDev, BOOL *pRising)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pRising	ユーザートリガー条件 (0=立ち下がり、 1 =立ち上がり)

ユーザートリガー条件に従って、ユーザートリガー情報がコールバック通知されます。
(ただし、ユーザートリガーモード= 0 ないし 2 の時のみに限ります。)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガー条件を取得します。

125) MvUsbCam_SetUserTrgChangeCB

DWORD MvUsbCam_SetUserTrgChangeCB(MVUSBCAM_UserTrgChangeCB lpFunc)

引数

lpFunc ユーザートリガーコールバック関数

ユーザートリガーコールバック関数の定義は、以下の通りです。
(MvUsbCam.h 参照のこと)

```
typedef void (CALLBACK* MVUSBCAM_UserTrgChangeCB)( DWORD, DWORD, BOOL );
```

また、ユーザートリガーコールバック関数の引数説明は、以下の通りです。

第1引数 デバイス番号

第2引数 ユーザートリガー検出回数

第3引数 トリガー検出時の I O レベル (0=LOW, 1=HIGH)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーのコールバック関数を指定します。

ユーザートリガーモニタ状態(*1)で、ユーザートリガー検知すると、
本関数で指定した関数にコールバックされます。

*** 1** ... MvUsbCam_StartUserTrgMonitor が C A L L されてから、
MvUsbCam_StopUserTrgMonitor が C A L L されるまでが、
ユーザートリガーモニター状態とします。

126) MvUsbCam_SetUserTrgMode

DWORD MvUsbCam_SetUserTrgMode(DWORD dwDev, int iDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat	ユーザートリガーモード 0=トリガーエッジ検出時に、コールバック通知。 1=ループ時間毎にコールバック通知。 2=トリガーエッジ検出時とループ時間毎にコールバック通知。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーモードを設定します。

本関数CALL直後から、ユーザートリガー動作が更新されます。

127) MvUsbCam_GetUserTrgMode

DWORD MvUsbCam_GetUserTrgMode(DWORD dwDev, int *pDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat	ユーザートリガーモード 0=トリガーエッジ検出時に、コールバック通知。 1=ループ時間毎にコールバック通知。 2=トリガーエッジ検出時とループ時間毎にコールバック通知。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーモード値を取得します。

128) MvUsbCam_SetUserTrgLoopTime

DWORD MvUsbCam_SetUserTrgLoopTime(DWORD dwDev, int iDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iDat	ユーザートリガーループ時間(msec)

この時間毎に、ユーザートリガー情報がコールバック通知されます。
(ただし、トリガーモード= 1 ないし 2 の時のみに限ります。)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーループ時間を設定します。

本関数CALL直後から、ユーザートリガー動作が更新されます。

129) MvUsbCam_GetUserTrgLoopTime

DWORD MvUsbCam_GetUserTrgLoopTime(DWORD dwDev, int *pDat)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
*pDat	ユーザートリガーループ時間(msec)

この時間毎に、ユーザートリガー情報がコールバック通知されます。
(ただし、トリガーモード= 1 ないし 2 の時のみに限ります。)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

ユーザートリガーループ時間を取得します。

130) MvUsbCam_SetPwmSetting

DWORD MvUsbCam_SetPwmSetting(DWORD dwDev, int iCh, DWORD dwHighWidth_Usec, DWORD dwLowWidth_Usec)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iCh MvUsbCam_PWM_CH_1 PWM1 のみ設定。
 MvUsbCam_PWM_CH_2 PWM2 のみ設定。
 MvUsbCam_PWM_CH_BOTH PWM1, PWM2 両方設定。
dwHighWidth_Usec High パルス幅(μ sec 単位)
dwLowWidth_Usec Low パルス幅(μ sec 単位)
iDat ホワイトポイント Y
 最大値、最小値は、MvUsbCam_GetParamInfo により取得できます。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE : Endpoint0 ライトエラー

説明

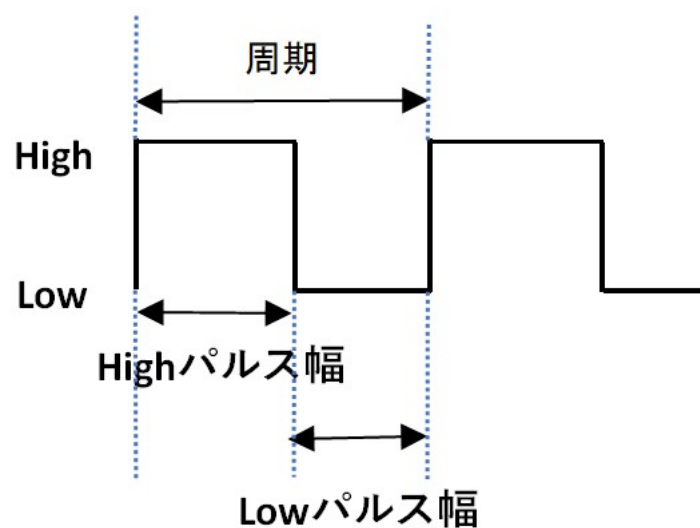
PWM(Pulse Width Modulation)設定のパルス幅を指定します。

本関数 CALL 後の PWM 出力値に影響します。

本関数は、GZ320/GZ303 専用関数です。

周期 = dwHighWidth_Usec + dwLowWidth_Usec

DUTY 比 = dwHighWidth_Usec ÷ 周期



131) MvUsbCam_GetPwmSetting

DWORD MvUsbCam_GetPwmSetting(DWORD dwDev, int iCh, DWORD *pHighWidth_Usec, DWORD *pLowWidth_Usec)

引数

dwDev デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0
iCh MvUsbCam_PWM_CH_1 PWM1 設定を取得。
 MvUsbCam_PWM_CH_2 PWM2 設定を取得。
*pHighWidth_Usec High パルス幅(μ sec 単位)
*pLowWidth_Usec Low パルス幅(μ sec 単位)

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO : デバイス番号が違います
MVUSBCAM_ERR_EP0_READ : Endpoint0 リードエラー

説明

PWM(Pulse Width Modulation)設定のパルス幅を指定します。
MvUsbCam_SetPwmSetting で、以前に設定した値を返します。

本関数は、GZ320／GZ303 専用関数です。

132) MvUsbCam_PwmStart

DWORD MvUsbCam_PwmStart(DWORD dwDev, int Mode)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0		
Mode	PWM 駆動モード		
	MvUsbCam_PWM_CH_1		PWM1 のみ動作。
	MvUsbCam_PWM_CH_2		PWM2 のみ動作。
	MvUsbCam_PWM_CH_BOTH		PWM1, PWM2 両方動作。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_WRITE	: Endpoint0 ライトエラー

説明

PWM(Pulse Width Modulation)の出力を開始します。

本関数は、GZ320／GZ303 専用関数です。

133) MvUsbCam_PwmStop

DWORD MvUsbCam_PwmStop(DWORD dwDev, int Mode)

引数

dwDev	デバイス番号(0 ～ MVUSBCAM_DEVICE_MAX - 1) ※通常は 0		
Mode	PWM 駆動モード		
	MvUsbCam_PWM_CH_1		PWM1 のみ停止。
	MvUsbCam_PWM_CH_2		PWM2 のみ停止。
	MvUsbCam_PWM_CH_BOTH		PWM1, PWM2 両方停止。

戻り値

正常終了の時は、MVUSBCAM_NO_ERROR を返します。それ以外は、以下の値などを返します。

MVUSBCAM_ERR_DEVICE_NO	: デバイス番号が違います
MVUSBCAM_ERR_EP0_READ	: Endpoint0 リードエラー

説明

PWM(Pulse Width Modulation)の出力を停止します。

本関数は、GZ320／GZ303 専用関数です。

134) MvUsbCam_ErrMsg

void MvUsbCam_ErrMsg(DWORD code)

引数

code エラーコード

戻り値

なし

説明

エラーコードに対応したメッセージをメッセージボックスで表示します。

135) MvUsbCam_GetErrText

void MvUsbCam_GetErrText(DWORD code, char* sMsg)

引数

code	エラーコード
*sMsg	エラーメッセージ

戻り値

なし

説明

エラーコードに対応したメッセージを取得します。

変更履歴

Date	Revision	Changes
Dec. 2015	0.90	New Release
May. 2016	1.00	MCM303 用関数を追加。
June.2016	1.01	2 関数の説明を追加。対応 OS を変更 (Windows 7 ～ 10 をサポート範囲とする。)
August.2016	1.02	SetLed, GetLed の説明修正。 ユーザーデータアクセス用関数を追加。
March.2017	1.03	TrgDelay 系関数を削除。 UserTrigger 系関数を追加。
Sept.2020	1.30	GZ3970 対応 トリガー系関数追加修正。
Jan. 2022	1.50	[3] 構造体の説明 修正。
May. 2022	1.60	GZ4120 対応関数を追加。
June. 2022	1.61	StrobeDelay 系関数の追加。
Jan. 2024	1.70	GZ320／GZ303 対応関数を追加。



〒950-0964 新潟市中央区網川原 2-44-13 3F

TEL : 025-282-7212 FAX : 025-282-7215

URL : <http://www.gazo.co.jp>