



株式会社マイクロビジョン

# MV 1 2 画像入力 ライブラリマニュアル

Ver.3.50

April 2010

株式会社マイクロビジョン

[目次]

[ 1 ] 動作環境.....	1
[ 2 ] 開発環境.....	1
[ 3 ] MV-1 2 ボードのインストール.....	1
[ 4 ] MV-1 2 ボード説明.....	2
[ 5 ] MV-1 2 画像入力用メモリ確保の説明.....	2
[ 6 ] カメラコンフィグレーションデータの説明.....	3
[ 7 ] MVC 1 2 _CAMCFG 構造体の説明.....	4
[ 8 ] MV 1 2 画像入力ライブラリ関数一覧.....	5
[ 9 ] MV 1 2 画像入力ライブラリ関数説明.....	7
1) Mvc12_Init.....	7
2) Mvc12_Init_Bd.....	8
3) Mvc12_Fini.....	9
4) Mvc12_GetBoardNum.....	10
5) Mvc12_GetBoardInfo.....	11
6) Mvc12_GetHDIB.....	12
7) Mvc12_SetImageNum.....	13
8) Mvc12_GetImageNum.....	14
9) Mvc12_Acquire.....	15
10) Mvc12_GetAcquireEvent.....	16
11) Mvc12_AcquireEndCheck.....	17
12) Mvc12_AcquireEnd.....	18
13) Mvc12_AcquireCancel.....	19
14) Mvc12_AcquireCancel2.....	20
15) Mvc12_HardwareReset.....	21
16) Mvc12_GetImageXsize.....	22
17) Mvc12_GetImageYsize.....	23
18) Mvc12_GetImageAddress.....	24
19) Mvc12_SetIniPath.....	25
20) Mvc12_GetIniPath.....	26
21) Mvc12_SetCameraConfig.....	27
22) Mvc12_SetCameraConfig2.....	28
23) Mvc12_LoadConfigFile.....	29
24) Mvc12_SaveConfigFile.....	30
25) Mvc12_LoadCameraConfig.....	31

26) Mvc12_GetCameraConfig .....	32
27) Mvc12_CameraConfigSetting .....	33
28) Mvc12_LoadShdData .....	34
29) Mvc12_SaveShdData .....	35
30) Mvc12_SetShdData .....	36
31) Mvc12_SetShdEnable .....	37
32) Mvc12_GetShdEnable .....	38
33) Mvc12_SetLsContinue .....	39
34) Mvc12_GetLsContinue .....	40
35) Mvc12_SetTrgMode .....	41
36) Mvc12_GetTrgMode .....	42
37) Mvc12_SetTrgExpsrTime .....	43
38) Mvc12_GetTrgExpsrTime .....	44
39) Mvc12_SetExpsrPls .....	45
40) Mvc12_GetExpsrPls .....	46
41) Mvc12_SetTrgPola .....	47
42) Mvc12_GetTrgPola .....	48
43) Mvc12_ConvertDib .....	49
44) Mvc12_IsNeedConvert .....	50
45) Mvc12_SetScanRate .....	51
46) Mvc12_GetScanRate .....	52
47) Mvc12_SetDspCommand .....	53
48) Mvc12_GetDspCommand .....	54
49) Mvc12_SetLutMode .....	55
50) Mvc12_GetLutMode .....	56
51) Mvc12_LoadLutData .....	57
52) Mvc12_SaveLutData .....	58
53) Mvc12_SetLutData .....	59
54) Mvc12_GetLutData .....	60
55) Mvc12_GetHist .....	61
56) Mvc12_GetHistValidArea .....	62
57) Mvc12_GetAcquireStopEvent .....	63
58) Mvc12_GetDmaPosition .....	64
59) Mvc12_GetImgValidStatus .....	65

## [ 1 ] 動作環境

PC 本体	: PC/AT 互換機
PCI チップセット	: Triton II 以上
CPU	: PentiumIII 850MHz 以上 (推奨 PentiumIV 1.4GHz 以上)
PCI-BIOS	: PCI-BIOS 2.1 以上
メモリ	: 128MBytes 以上 (推奨 256MBytes 以上)
対応 OS	: Microsoft Windows 2000 / XP : Microsoft Windows Vista 32bit : Microsoft Windows 7 32bit / 64bit

## [ 2 ] 開発環境

●コンパイラ                   Microsoft Visual Studio 2005 以降

●ライブラリファイル

MV 1 2 画像入力ライブラリ	……Mvc12.lib
上記ライブラリのヘッダファイル	……Mvc12.h
上記ライブラリの DLL ファイル	……Mvc12.dll
MV 1 2 メモリ確保プログラム	……Mvc_Mem12.exe

## [ 3 ] MV-1 2 ボードのインストール

MV-1 2 基板と、そのドライバのインストールについては、画像マネージャーのマニュアルを参照してください。

## [4] MV-12ボード説明

MV-12の仕様は、以下の通りです。

仕様バス	PCIバス
入力信号	RS422、LVDS 32/24/16/8Bitデジタル
画像レート	外部クロック/内部クロック 40/20/10/5MHz
同期信号	エリアセンサ 外部同期入力 HD VD CLOCK ラインセンサ 内部同期入力 START CLOCK
画像サイズ	水平方向 4~5150ドット 垂直方向 4~16000ライン
画像メモリ	8Mバイト × 2面
輝度調整	シェーディング機能 (ラインセンサ)
取込信号	内部モード 外部モード (トリガー信号、エンコーダー信号)
外形寸法	106.88mm × 174.63mm
カメラ	エリアセンサカメラ・・・140万画素に対応 ラインセンサカメラ・・・5150万画素まで対応
インタフェース	MV12-DSP インタフェースボード接続によりTI社DSPボード TMS320C6201/TMS320C6207と接続可能
取込モード	1フレームモード 連続フレームモード

## [5] MV-12画像入力用メモリ確保の説明

画像入力のバッファサイズを変更したい場合は、**FrmMem.exe** を起動して下さい。  
(**FrmMem.exe** は画像マネージャーインストール時にフォルダを変更していなければ、**C:\¥Mvc** フォルダ内にあります。**mvc\_mem12.exe** と同じフォルダに置いてください。)

**Mvc12\_Acquire** を実行すると、画像入力バッファに入力イメージが格納されます。

画像入力バッファにアクセスするには、**Mvc12\_GetImageAddress** を使用します。

**Mvc12\_GetImageAddress** は、画像の横方向の先頭アドレスを取得する関数です。  
ただし、**Mvc12\_SetImageNum** で指定したフレーム番号により **Mvc12\_GetImageAddress** で取得できるイメージデータは異なってきます。

先頭アドレスから **MVC12\_CAMCFG** 構造体の「**dwPixBit**」で指定した画素単位でデータが並んでいます。

## [ 6 ] カメラコンフィグレーションデータの説明

カメラコンフィグレーションデータのファイル名は、「カメラ名.dat」になっています。  
ファイルフォーマットは以下の様になっています。

### [MVC12\_CAM\_CFG]

Sign . . . . . 「Mvc12CamCfg」 固定  
 Comment . . . . . コメント (無くても問題ありません)  
 Xsize . . . . . 画像の X サイズ (ピクセル単位)  
 Ysize . . . . . 画像の Y サイズ (ピクセル単位)  
 FrameNum . . . . . 画像フレーム数 (1 以上)  
     例) 1300×1000 のエリアセンサカメラの場合  
         1 フレームが 1.3Mbyte です。  
         メモリ確保サイズを多くすればフレーム数を増やすことができます。

PixBit . . . . . 1 画素の Bit 数  
 Color . . . . . カラー／白黒カメラの指定 (0:白黒 1:RGB)  
 PixType . . . . . 画素タイプ (0:32Bit 1:8Bit 2:10Bit 3:Shift2\_8Bit 4:16Bit)  
 InputBit . . . . . カメラから MV 1 2 への入力ビット数を指定 (LUT 変換時に使用)  
 DotNum . . . . . ドット数 (通常は X サイズと同じ。デュアル出力時のみ X サイズ/2)  
 DivPulse . . . . . エンコーダの分周パルス  
     (ラインセンサカメラ使用時、かつ、エンコーダありの時有効)  
 Digital . . . . . カメラ種別 (0:ラインセンサ 1:エリアセンサ)  
 WrDelay . . . . . FIFO WRITE DELAY DOT の指定  
 RdDelay . . . . . FIFO READ DELAY DOT の指定  
 Clk . . . . . 画像入力クロック  
     (0:DCLK 1:80M 2:40M 3:20M 4:10M 5:5M 6:DCLK 逆位相)  
 Irq . . . . . 割り込みの指定 (0:なし 1:あり)  
 VPls . . . . . 割り込みパルス幅 (ラインセンサカメラ専用)  
 ScanRate . . . . . スキャンレート (ラインセンサカメラ専用)  
     ※ScanRate > DotNum + WrDelay に設定してください。  
 Interlace . . . . . インターレースモード (0: なし 1: あり) (エリアセンサカメラ専用)  
 RsvOut0 . . . . . カメラケーブル出力端子 RSVOUT0 の出力論理  
 RsvOut1 . . . . . カメラケーブル出力端子 RSVOUT1 の出力論理  
 RsvOut2 . . . . . カメラケーブル出力端子 RSVOUT2 の出力論理  
 FifoWrSel . . . . . FIFO WRITE SELECT  
     (0:従来 FifoWrDelay 1:D\_HD 2:RSVIN1 3:RSVIN2 使用)  
 Convert . . . . . 画像変換形式指定( 1:CA-D6 等 4 ヶpp 変換 3:Interlace 変換 ) ※2  
 SyncPola . . . . . 同期信号極性  
 Wen . . . . . 1:WEN 信号使用 ... 現在 MegaScope の ResetStart モードで使用 ※2  
 StopInt . . . . . 1: 画像入力中止割り込みを使用

※2 但し、Interlace モードは、まだサポートしておりません。

## [ 7 ] MVC 1 2 \_CAMCFG 構造体の説明

MVC12\_CAMCFG 構造体は、以下の通りです。

(Mvc12\_LoadCameraConfig 関数、Mvc12\_GetCameraConfig 関数で使用します。)

szComment	カメラコンフィグ内容説明。
VerNum	バージョン情報
dwXsize	画像サイズ X(横方向)を指定。
dwYsize	画像サイズ Y(縦方向)を指定。
dwFrameNum	画像フレーム数の指定。
dwPixBit	1 画素の Bit 数
dwColor	カラー／白黒カメラの指定 (0:白黒 1:RGB)
dwPixType	画素タイプを指定 (0:32Bit 1:8Bit 2:10Bit 3:8Bit[2Bit 右シフトあり] 4:16Bit)
dwInputBit	カメラからMV 1 2 への入力ビット数を指定 (LUT 変換時に使用)
dwDotNum	ドット数の指定。 (通常は dwXsize と同じ。デュアル出力時のみ dwXsize / 2)
dwDivPulse	エンコーダの分周パルスを指定。 (ラインセンサ使用時、かつ、エンコーダありの時有効)
dwDigital	エリア／ラインセンサの指定。 (1:エリアセンサ 0:ラインセンサ)
dwWrDelay	FIFO WRITE DELAY DOT の指定。
dwRdDelay	FIFO READ DELAY DOT の指定。
dwClk	画像入力クロックの指定。 (0:DCLK 1:80MHz 2:40MHz 3:20MHz 4:10MHz 5:5MHz 6:DCLK REVERSE)
dwIrq	割り込みの指定。 (0:なし, 1:あり)
dwVPls	割り込みパルス幅の指定。初期設定 30。[ラインセンサ専用]
dwScanRate	スキャンレート [ラインセンサ専用] (dwScanRate > dwDotNum + dwWrDelay に設定してください)
dwInterlace	インターレースモードの指定。 (0:なし, 1:あり)
dwRsvOut0	カメラケーブル出力端子 RSVOUT0 の出力論理
dwRsvOut1	カメラケーブル出力端子 RSVOUT1 の出力論理
dwRsvOut2	カメラケーブル出力端子 RSVOUT2 の出力論理
dwFifoWrSel	FIFO WRITE SELECT (0:従来 FifoWrDelay 1:D_HD 2:RSVIN1 3:RSVIN2 使用)
dwConvert	画像変換形式指定(1:CA-D6 等 4タップ 変換)
dwSyncPola	同期信号極性
dwWen	WEN 信号 (0:なし, 1:あり)
StopInt	画像入力中止割り込み (0:なし, 1:あり)
dwReverve	予約パラメータ
dwReverve1	予約パラメータ
dwReverve2	予約パラメータ
dwReverve3	予約パラメータ
dwReverve4	予約パラメータ
dwReverve5	予約パラメータ

## [ 8 ] MV 1 2 画像入力ライブラリ関数一覧

MV 1 2 画像入力ライブラリには、以下の関数があります。

1) Mvc12_Init	MV 1 2 ボードの初期化处理
2) Mvc12_Init_Bd	MV 1 2 ボードの初期化处理 (ボード指定)
3) Mvc12_Fini	MV 1 2 ボードの終了処理
4) Mvc12_GetHDIB	入力画像の取得
5) Mvc12_GetBoardNum	MV 1 2 ボード数の取得
6) Mvc12_GetBoardInfo	MV 1 2 ボード情報の取得
7) Mvc12_SetImageNum	フレーム番号の設定
8) Mvc12_GetImageNum	フレーム番号の取得
9) Mvc12_Acquire	画像入力の実行
10) Mvc12_GetAcquireEvent	画像入力終了の為の同期イベントの取得
11) Mvc12_AcquireEndCheck	画像入力の終了チェック
12) Mvc12_AcquireEnd	画像入力の終了処理
13) Mvc12_AcquireCancel	画像入力のキャンセル処理
14) Mvc12_AcquireCancel2	画像入力のキャンセル処理 (戻り値あり)
15) Mvc12_HardwareReset	ハードウェアのリセット
16) Mvc12_GetImageXsize	画像入力Xサイズの取得
17) Mvc12_GetImageYsize	画像入力Yサイズの取得
18) Mvc12_GetImageAddress	画像入力アドレスの取得
19) Mvc12_SetIniPath	Mvc12.ini のパス設定
20) Mvc12_GetIniPath	Mvc12.ini のパス取得
21) Mvc12_SetCameraConfig	カメラコンフィグレーションの設定
22) Mvc12_SetCameraConfig2	カメラコンフィグレーションの設定 (画像サイズ)
23) Mvc12_LoadConfigFile	カメラコンフィグレーションファイルの読込
24) Mvc12_SaveConfigFile	カメラコンフィグレーションファイルの保存
25) Mvc12_LoadCameraConfig	カメラコンフィグレーションファイルの読込
26) Mvc12_GetCameraConfig	カメラコンフィグレーションデータの取得
27) Mvc12_CameraConfigSetting	カメラコンフィグレーションデータの表示
28) Mvc12_LoadShdData	シェーディングデータの読込
29) Mvc12_SaveShdData	シェーディングデータの保存
30) Mvc12_SetShdData	シェーディング調整
31) Mvc12_SetShdEnable	シェーディング調整の有効/無効の設定
32) Mvc12_GetShdEnable	シェーディング調整の有効/無効の取得
33) Mvc12_SetLsContinue	ラインセンサカメラ入力モードの設定
34) Mvc12_GetLsContinue	ラインセンサカメラ入力モードの取得
35) Mvc12_SetTrgMode	トリガーモードの設定
36) Mvc12_GetTrgMode	トリガーモードの取得
37) Mvc12_SetTrgExpsrTime	エリアセンサ用の露光時間の設定
38) Mvc12_GetTrgExpsrTime	エリアセンサ用の露光時間の取得
39) Mvc12_SetExpsrPls	ラインセンサ用の露光時間の設定
40) Mvc12_GetExpsrPls	ラインセンサ用の露光時間の取得
41) Mvc12_SetTrgPola	トリガー極性の設定
42) Mvc12_GetTrgPola	トリガー極性の取得

43) Mvc12_ConvertDib	画像変換( 入力イメージの復元のためのもの )
44) Mvc12_IsNeedConvert	画像変換が必要かどうかの確認
45) Mvc12_SetScanRate	スキャンレートの設定
46) Mvc12_GetScanRate	スキャンレートの取得
47) Mvc12_SetDspCommand	D S P コマンドの設定
48) Mvc12_GetDspCommand	D S P コマンドの取得
49) Mvc12_SetLutMode	画像入力時のデータ変換モード指定
50) Mvc12_GetLutMode	画像入力時のデータ変換モード取得
51) Mvc12_LoadLutData	L U T 調整データファイルの読込
52) Mvc12_SaveLutData	L U T 変換用データファイルの保存
53) Mvc12_SetLutData	L U T 変換用データの設定
54) Mvc12_GetLutData	L U T 変換用データの取得
55) Mvc12_GetHist	ヒストグラムの取得
56) Mvc12_GetHistValidArea	ヒストグラム有効領域の計算
57) Mvc12_GetAcquireStopEvent	入力中止割り込み用イベントの取得
58) Mvc12_GetDmaPosition	DMA 転送位置の取得
59) Mvc12_GetImgValidStatus	画像入力状態の取得

## [ 9 ] MV 1 2 画像入力ライブラリ関数説明

MV 1 2 画像入力ライブラリの関数を詳しく説明します。

\*\*\*\*\*

### 1) Mvc12\_Init

```
int Mvc12_Init (void)
```

#### 引数

なし

#### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

1: MV-12 ボードがありません

2: Mvc12.sys がありません

3: メモリ確保ミス

4: リソースエラー (MV-1 2 ボードのリソースなし)

7: 起動済みエラー

-1: キーがありません

-2: キーが異なります

#### 説明

MV-1 2 ボードを使用するための初期化処理を実行します。

例) アプリケーションでMV-1 2を使用するための初期化関数です。MV-1 2を使用する場合は、アプリケーション起動時に必ず記述して下さい。

```
BOOL CMVcApp::InitInstance()
```

```
{
    // 画像入力ライブラリ初期化
    int iErr = Mvc12_Init();
    if (iErr){
        AfxMessageBox("MV-12 エラー", MB_OK);
        return FALSE;
    }
    return TRUE;
}
```

\*\*\*\*\*

## 2) Mvc12\_Init\_Bd

```
void Mvc12_Init_Bd ( DWORD dwBd )
```

### 引数

なし

### 戻り値

なし

### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

1: MV-12 ボードがありません

2: Mvc12.sys がありません

3: メモリ確保ミス

4: リソースエラー (MV-1 2 ボードのリソースなし)

6: ボード番号エラー

7: 起動済みエラー

-1: キーがありません

-2: キーが異なります

### 説明

MV-1 2 ボードを使用するための初期化処理を実行します。

例) アプリケーションでMV-1 2を使用するための初期化関数です。MV-1 2を使用する場合は、アプリケーション起動時に必ず記述して下さい。

```
BOOL CMVcApp::InitInstance()
```

```
{
    // 画像入力ライブラリ初期化
    int iErr = Mvc12_Init_Bd(0);    // 1枚目のボード
    if ( iErr ){
        AfxMessageBox("MV-12 エラー", MB_OK);
        return FALSE;
    }
    return TRUE;
}
```

\*\*\*\*\*

3) Mvc12\_Fini  
void Mvc12\_Fini (void)

引数

なし

戻り値

なし

説明

MV-1 2 ボードの終了処理を実行します。

例) アプリケーションでMV-1 2 を終了するための関数です。アプリケーション終了時に必ず記述して下さい。

```
int CMVcApp::ExitInstance()
{
    // 画像入力ライブラリ終了
    Mvc12_Fini();

    return CWinApp::ExitInstance();
}
```

\*\*\*\*\*

4) Mvc12\_GetBoardNum  
int Mvc12\_GetBoardNum ( DWORD \*pNum )

引数

\*pNum MV-12 のボード数を返します。

戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

1: ボードがありません

説明

使用している MV-12 ボード数を取得します。

\*\*\*\*\*

### 5) Mvc12\_GetBoardInfo

```
void Mvc12_GetBoardInfo ( MV12_BOARD_INFO *pInfo )
```

#### 引数

\*pInfo ボード情報を返します。

MV21\_BOARD\_INFO は、Mvc12.h で、次のように定義してあります。

```
typedef struct
{
    DWORD dwBus;           // バス番号
    DWORD dwDev;          // デバイス番号
    DWORD dwFunc;         // ファンクション番号
} MV12_BOARD_INFO
```

#### 戻り値

なし

#### 説明

MV-12 のボード情報を取得します。

MV-12 ライブラリ初期化関数( Mvc12\_Init、Mvc12\_Init\_Bd )を CALL した後、本関数を CALL してください。

\*\*\*\*\*

6) Mvc12\_GetHDIB  
HANDLE Mvc12\_GetHDIB (void)

引数  
なし

戻り値  
画像データのハンドル値を返します。(0 はエラー)

説明  
Mvc12\_SetImageNum で設定したフレーム番号の画像ハンドル値を取得します。

例)  
画像取得例です

```
#include "Mvc12.h"

HANDLE *phDib;
MVC12_CAMCFG stCamCfg;

Mvc12_GetCameraConfig( &stCamCfg );

phDib = new HANDLE[ stCamCfg.dwFrameNum ];

for (DWORD i = 0; i < stCamCfg.dwFramNum; i++)
{
    Mvc12_SetImageNum( (int)i );
    phDib[i] = Mvc12_GetHDIB();
}

.....
.....

delete []phDib;
```

\*\*\*\*\*

7) Mvc12\_SetImageNum

```
int Mvc12_SetImageNum (int iImgNum);
```

引数

iImgNum 画像フレーム番号 (0 ~ dwFrameNum- 1)

戻り値

設定前の画像フレーム番号を取得します。

※dwFrameNum はカメラコンフィグデータの MVC12\_CAMCFG 構造体で設定されています。

説明

画像フレーム番号を設定します。

例)

```
#include "Mvc12.h"

HANDLE *phDib;
MVC12_CAMCFG stCamCfg;

Mvc12_GetCameraConfig( &stCamCfg );

phDib = new HANDLE[ stCamCfg.dwFrameNum ];

for (DWORD i = 0; i < stCamCfg.dwFramNum; i++)
{
    Mvc12_SetImageNum( (int)i );
    phDib[i] = Mvc12_GetHDIB();
}

.....

delete []phDib;
```

\*\*\*\*\*

8) Mvc12\_GetImageNum  
int Mvc12\_GetImageNum (void);

引数  
なし

戻り値  
iImgNum 画像フレーム番号 (0 ~ dwFrameNum-1)

説明  
画像フレーム番号を取得します。

※dwFrameNum はカメラコンフィグデータの MVC12\_CAMCFG 構造体で設定されています。

例)

```
#include "Mvc12.h"

int i;
HANDLE *phDib;
MVC12_CAMCFG stCamCfg;

Mvc12_GetCameraConfig( &stCamCfg );

phDib = new HANDLE[ stCamCfg.dwFrameNum ];

i = Mvc12_GetImageNum();
Mvc12_SetHDIB(phDib[i]);

.....
.....
```

\*\*\*\*\*

### 9) Mvc12\_Acquire

```
int Mvc12_Acquire (int iMultiFrm, int iStartFrm, int iLoopNum,
                  int iWait, int iTimeout, int iTimeWait )
```

#### 引数

iMultiFrm	フレーム順指定 (iLoopNum = 0 の時、有効) 0:同じフレームに入力されます。 1:入力されるフレーム番号が 0 ~ dwFrameNum-1 の順序で変わります。
iStartFrm	開始フレーム (0 ~ dwFrameNum-1)
iLoopNum	入力処理回数 0:連続入力を行います。 0 以外 : 設定した回数だけ入力処理を行います。
iWait	画像待機 0:画像入力の終了を待ちません。 1:画像入力の終了まで待ちます。
iTimeout	入力オーバータイム (msec) ※割り込み使用時、有効
iTimeWait	入力終了待ち時間 (msec) ※割り込み非使用時、有効

#### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

- 1:パラメータエラー
- 2:タイムオーバー (iWait = 1 の時のみ)
- 3:その他のエラー

#### 説明

MV 1 2 ボードから画像入力を実行します。

※dwFrameNum はカメラコンフィグデータの MVC12\_CAMCFG 構造体で設定されています。

\*\*\*\*\*

10) Mvc12\_GetAcquireEvent  
 int Mvc12\_GetAcquireEvent (HANDLE\* pObject)

引数

\*pObject      同期イベント

戻り値

0 固定

説明

画像入力終了のための同期イベントを取得します。  
 このイベントは、ライブラリで管理しているため、ユーザーでハンドルを解放する必要はありません。

例)

画像入力例です。

```
#include "Mvc12.h"

DWORD dwRet;
HANDLE hAcqEnd;

Mvc12_GetAcquireEvent (&hAcqEnd);

Mvc12_Acquire(0, 0, 1, 0, 1000, 1000);

dwRet = WaitForSingleObject( hAcqEnd, 1000);
if (dwRet == WAIT_OBJECT_0)
{
    // 入力終了
    ::ResetEvent(hAcqEnd);
}
else
{
    // 画像入力異常 or タイムオーバー
}
```

\*\*\*\*\*

11) Mvc12\_AcquireEndCheck  
int Mvc12\_AcquireEndCheck (void)

引数

なし

戻り値

- 0:画像入力終了しました
- 1:画像入力がまだ終了していません
- 2:タイムアウト
- 3:途中終了
- 4:その他のエラー

説明

MV 1 2 ボードから画像入力の終了確認を行います。

例)

画像入力例です。

```
#include "Mvc12.h"

.....
.....
Mvc12_Acquire(0, 0, 1, 0, 1000, 1000);

int iRet;
while(1){
    // 画像入力終了後ループを抜けます
    iRet = Mvc12_AcquireEndCheck();
    if (iRet == 0)
        break;
}
.....
.....
```

\*\*\*\*\*

12) Mvc12\_AcquireEnd  
 int Mvc12\_AcquireEnd (int iTimeout)

引数

iTimeout      入力オーバータイム (msec) ※割り込み使用時、有効

戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

- 2:タイムアウト
- 3:途中終了
- 4:その他のエラー

説明

画像入力後の後処理を行います。

Mvc12\_AcquireEndCheck を Call し、戻り値が 0 の場合はこの関数を呼ぶ必要はありません。

画像入力処理が完全に終了していない状態で、カメラからの映像出力時間に比べて iTimeout の値が小さい場合、この関数を呼ぶと入力画像が不完全になる可能性があります。

例)

画像入力例です。

```
#include "Mvc12.h"

.....

.....

Mvc12_Acquire(0, 0, 1, 0, 1000, 1000);

// 他の処理

Mvc12_AcquireEnd(200);            // 画像入力終了

.....

.....
```

\*\*\*\*\*

13) Mvc12\_AcquireCancel  
void Mvc12\_AcquireCancel (void)

引数

なし

戻り値

なし

説明

画像入力をキャンセルします。

Mvc12\_AcquireEndCheck を Call し、戻り値が 0 の場合はこの関数を呼ぶ必要はありません。

画像入力処理が完全に終了していない状態で、この関数を呼ぶと入力画像が不完全になる可能性があります。

例)

画像入力例です。

```
#include "Mvc12.h"

.....
.....
Mvc12_Acquire(0, 0, 1, 0, 1000, 1000);

// 他の処理

Mvc12_AcquireCancel ();           // 画像入力キャンセル
.....
.....
```

\*\*\*\*\*

14) Mvc12\_AcquireCancel2  
 int Mvc12\_AcquireCancel2 (void)

引数

なし

戻り値

DMA ABORT 処理のステータス( 0 : 正常終了、 1 : 異常終了 )

説明

画像入力をキャンセルします。

Mvc12\_AcquireEndCheck を Call し、戻り値が 0 の場合はこの関数を呼ぶ必要はありません。

画像入力処理が完全に終了していない状態で、この関数を呼ぶと入力画像が不完全になる可能性があります。

本関数の戻り値が 1 の場合は、MV 1 2 のハードウェアリセットを実行するために、Mvc12\_HardwareReset を CALL する必要があります。

例)

画像入力例です。

```
#include "Mvc12.h"

.....
.....
Mvc12_Acquire(0, 0, 1, 0, 1000, 1000);

// 他の処理

if ( Mvc12_AcquireCancel () )    // 画像入力キャンセル
{
    // DMA Abort 失敗
    Mvc12_HardwareReset();
    Sleep(300);
}

.....
.....
```

\*\*\*\*\*

15) Mvc12\_HardwareReset  
void Mvc12\_ HardwareReset( void )

引数  
なし

戻り値  
なし

説明  
MV 2 2 のハードウェアリセットを実行します。

\*\*\*\*\*

16) Mvc12\_GetImageXsize  
int Mvc12\_GetImageXsize (void)

引数  
なし

戻り値  
入力画像の横 (X) サイズを取得します。

説明  
MV 1 2 ボードからの入力された画像の横サイズを取得します。

例)  
画像入力例です

```
#include "Mvc12.h"  
  
.....  
.....  
// 入力画像サイズ設定  
int iImgX = Mvc12_GetImageXsize();  
int iImgY = Mvc12_GetImageYsize();  
.....  
.....
```

\*\*\*\*\*

17) Mvc12\_GetImageYsize  
int Mvc12\_GetImageYsize (void)

引数  
なし

戻り値  
入力画像の縦 (Y) サイズを取得します。

説明  
MV 1 2 ボードからの入力された画像の縦サイズを取得します。

例)  
画像入力例です

```
#include "Mvc12.h"  
  
.....  
.....  
// 入力画像サイズ設定  
int iImgX = Mvc12_GetImageXsize();  
int iImgY = Mvc12_GetImageYsize();  
.....  
.....
```

\*\*\*\*\*

18) Mvc12\_GetImageAddress  
LPVOID Mvc12\_GetImageAddress (int y)

引数

y 入力画像の横ライン行

戻り値

指定ラインの画像格納アドレスを返します。

説明

MV 1 2 ボードの入力画像メモリのアドレスを取得します。

画像入力処理終了後、このメモリアドレスに新しい画像が格納されます。

Mvc12\_SetImageNum で指定した画像フレーム番号のアドレスを取得します。

\*\*\*\*\*

### 19) Mvc12\_SetIniPath

BOOL Mvc12\_SetIniPath ( cher \*path )

#### 引数

path      IniFile のパス

#### 戻り値

正常終了時は、FALSE 以外を返します。

#### 説明

Mvc12.ini のパスを設定します。

主に Visual Basic 用に作成された関数です。

初期状態は、実行ファイルのパスになっていますが、本関数でパスを変更することが可能です。

特に、Visual Basic のデザインモードでは、Visual Basic(VBS ext 等)が実行ファイルのパスになるため、本関数でパスを変更する必要があります。

なお、本関数は、関数 (Mvc12\_Init) の前に Call してください。

\*\*\*\*\*

## 20) Mvc12\_GetIniPath

```
void Mvc12_GetIniPath ( char *path )
```

### 引数

path      IniFile のパス

### 戻り値

なし

### 説明

Mvc12.ini のパスを取得します。  
主に Visual Basic 用に作成された関数です。

\*\*\*\*\*

## 21) Mvc12\_SetCameraConfig

int Mvc12\_SetCameraConfig (MVC12\_CAMCFG\* lpstCamCfg)

### 引数

\*lpstCamCfg      カメラコンフィグ構造体

### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

1:パラメータエラー

4:確保メモリ不足

### 説明

カメラコンフィグ構造体指定で、入力環境を設定します。

\*\*\*\*\*

## 22) Mvc12\_SetCameraConfig2

```
int Mvc12_SetCameraConfig2(MVC12_CAMCFG* lpstCamCfg)
```

### 引数

\*lpstCamCfg      カメラコンフィグ構造体

### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

1:パラメータエラー

4:確保メモリ不足

### 説明

カメラコンフィグ構造体指定で、入力環境を設定します。

※画像サイズのみを変更する場合に使用してください。

\*\*\*\*\*

### 23) Mvc12\_LoadConfigFile

int Mvc12\_LoadConfigFile (char\* szFullPath, MVC12\_CAMCFG\* lpstCamCfg)

#### 引数

\*szFullPath        カメラコンフィグレーションファイルの絶対パス  
\*lpstCamCfg        カメラコンフィグ構造体

#### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

- 1:パラメータエラー
- 2:ファイルがありません
- 3:ファイル形式が違います

#### 説明

カメラコンフィグレーションファイルを読み込み、データを構造体に設定します。

\*\*\*\*\*

#### 24) Mvc12\_SaveConfigFile

int Mvc12\_SaveConfigFile (char\* szFullPath, MVC12\_CAMCFG\* lpstCamCfg)

#### 引数

\*szFullPath        カメラコンフィグレーションファイルの絶対パス  
\*lpstCamCfg        カメラコンフィグ構造体

#### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

- 1:パラメータエラー
- 2:ファイルがありません

#### 説明

カメラコンフィグ構造体のデータをファイルに保存します。

\*\*\*\*\*

25) Mvc12\_LoadCameraConfig  
int Mvc12\_LoadCameraConfig (char\* szFullPath)

引数

\*szFullPath           カメラコンフィグレーションファイルの絶対パス

戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

1:パラメータエラー

(dwXsize、dwYsize、dwFrameNum のサイズを満たすメモリが確保されていない場合)

2:ファイルがありません

3:ファイル形式が違います

説明

カメラコンフィグレーションファイルを読み込みます。

※dwXsize、dwYsize、dwFrameNum はカメラコンフィグデータの MVC12\_CAMCFG 構造体で設定されています。

\*\*\*\*\*

26) Mvc12\_GetCameraConfig

int Mvc12\_GetCameraConfig (MVC12\_CAMCFG\* lpstCamCfg)

引数

\*lpstCamCfg カメラコンフィグレーションのパラメータ  
MVC12\_CAMCFG 構造体へのポインタ

戻り値

0 固定

説明

カメラコンフィグレーションデータを取得します。

\*\*\*\*\*

27) Mvc12\_CameraConfigSetting  
void Mvc12\_CameraConfigSetting (void)

引数

なし

戻り値

なし

説明

現在ロードされているカメラコンフィグレーションファイルの内容を表示します。

\*\*\*\*\*

28) Mvc12\_LoadShdData  
int Mvc12\_LoadShdData (char \*szFullPath)

引数

\*szFullPath シェーディングデータファイルの絶対パス

戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

- 1:パラメータエラー
- 2:ファイルがありません
- 3:ファイル形式が違います
- 4:内部メモリ確保エラー

説明

シェーディングデータファイルを読み込みます。

\*\*\*\*\*

## 29) Mvc12\_SaveShdData

int Mvc12\_SaveShdData (char \*szFullPath)

### 引数

\*szFullPath シェーディングデータファイルの絶対パス

### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。

- 1:パラメータエラー
- 2:ファイルがありません
- 3:ファイル形式が違います
- 4:シェーディング用メモリ確保エラー

### 説明

シェーディングデータファイルを保存します。

\*\*\*\*\*

### 30) Mvc12\_SetShdData

int Mvc12\_SetShdData (int y\_line, int a, int b, int c)

#### 引数

y_line	行
a	(未使用)
b	(未使用)
c	(未使用)

#### 戻り値

正常終了の時は、0 を返します。

#### 説明

Mvc12\_SetImageNum で設定されているフレームの y\_line 行目の画像データを元にシェーディングデータを設定します。

\*\*\*\*\*

31) Mvc12\_SetShdEnable

void Mvc12\_SetShdEnable (BOOL bValid)

引数

bValid        FALSE 以外:有効  
              FALSE:無効

戻り値

なし

説明

シェーディング調整の有効/無効を設定します。

\*\*\*\*\*

32) Mvc12\_GetShdEnable

void Mvc12\_GetShdEnable (BOOL \*bValid)

引数

\*bValid        FALSE 以外:有効  
                  FALSE:無効

戻り値

なし

説明

シェーディング調整の有効/無効を取得します。

\*\*\*\*\*

### 33) Mvc12\_SetLsContinue

void Mvc12\_SetLsContinue (BOOL bContinue)

#### 引数

bContinue    FALSE 以外:連続入力モード  
              FALSE:1SHOT モード

#### 戻り値

なし

#### 説明

ラインセンサカメラの入力モードを設定します。

Mvc12\_SetTrgMode 関数を実行する前に、本関数を実行してください。

\*\*\*\*\*

34) Mvc12\_GetLsContinue  
void Mvc12\_GetLsContinue (BOOL\* bContinue)

引数

\*bContinue FALSE 以外:連続入力モード  
FALSE:1SHOT モード

戻り値

なし

説明

ラインセンサカメラの入力モードを取得します。

\*\*\*\*\*

### 35) Mvc12\_SetTrgMode

```
void Mvc12_SetTrgMode (int iAsTrg, BOOL bAsContinue, int iLsTrg)
```

#### 引数

iAsTrg	エリアセンサ用パラメータ 0 : トリガーなし 1 : 外部トリガー 2 : 外部トリガー (VINIT 出力) 3 : 内部トリガー (VINIT 出力)
bAsContinue	エリアセンサ用パラメータ 0 : 1 SHOT モード 1 : 連続入力モード (iAsTrg = 1 の時のみ有効)
iLsTrg	ラインセンサ用パラメータ 0 : トリガーなし 1 : 外部トリガー 2 : エンコーダトリガー

#### 戻り値

なし

#### 説明

トリガー状態を設定します。  
 画像入力中には設定を行わないでください。

\*\*\*\*\*

36) Mvc12\_GetTrgMode

void Mvc12\_GetTrgMode (int \*iAsTrg, BOOL \*bAsContinue, int \*iLsTrg)

引数

*iAsTrg	エリアセンサ用パラメータ 0 : トリガーなし 1 : 外部トリガー 2 : 外部トリガー (トリガー出力あり) 3 : 内部トリガー (トリガー出力あり)
*bAsContinue	エリアセンサ用パラメータ 0 : 1 SHOT モード 1 : 連続入力モード
*iLsTrg	ラインセンサ用パラメータ 0 : トリガーなし 1 : 外部トリガー 2 : エンコーダトリガー

戻り値

なし

説明

トリガー状態を取得します。

\*\*\*\*\*

### 37) Mvc12\_SetTrgExpsrTime

```
void Mvc12_SetTrgExpsrTime (int mode, int Tim)
```

#### 引数

mode	単位
	0 : $\mu$ sec
	1 : msec
Tim	露光時間

#### 戻り値

なし

#### 説明

トリガー有効時の露光時間を設定します。  
設定後の画像入力から影響します。

※エリアセンサカメラ専用

\*\*\*\*\*

38) Mvc12\_GetTrgExpsrTime  
void Mvc12\_GetTrgExpsrTime (int \*mode, int \*Tim)

引数

*mode	単位
	0 : $\mu$ sec
	1 : msec
*Tim	露光時間

戻り値

なし

説明

現在設定されているトリガー有効時の露光時間を取得します。

※エリアセンサカメラ専用

\*\*\*\*\*

39) Mvc12\_SetExpsrPls  
int Mvc12\_SetExpsrPls (int clk)

引数

clk       露光時間 (クロック)

戻り値

以前の設定値を返します。

説明

トリガー有効時の露光時間を設定します。  
設定後の画像入力から影響します。

※ラインセンサカメラ専用

\*\*\*\*\*

40) Mvc12\_GetExpsrPls  
int Mvc12\_GetExpsrPls (void)

引数

なし

戻り値

露光時間（クロック）を返します。

説明

現在設定されているトリガー有効時の露光時間を取得します。

※ラインセンサカメラ専用

\*\*\*\*\*

41) Mvc12\_SetTrgPola  
int Mvc12\_SetTrgPola (int mode)

引数

mode     トリガー極性  
          0:負  
          1:正

戻り値

以前の設定値を返します。

説明

トリガー極性を設定します。  
初期値は 1:正です。  
設定後の画像入力から影響します。  
外部トリガー未使用の場合は、1:正して下さい。

\*\*\*\*\*

42) Mvc12\_GetTrgPola  
int Mvc12\_GetTrgPola (void)

引数

なし

戻り値

トリガー極性を返します。

0:負

1:正

説明

現在設定されているトリガー極性を取得します。

\*\*\*\*\*

#### 43) Mvc12\_ConvertDib

int Mvc12\_ConvertDib (HANDLE hDst, HANDLE hSrc, int mode)

#### 引数

hDst 変換先の DIB ハンドル  
hSrc 変換元の DIB ハンドル  
mode 1:DALSA 4 タップ入力の変換 (CA-D6 用)  
2: DALSA 4 タップ入力の変換 (CT-E4 用)  
3:Interlace 変換

#### 戻り値

正常終了の時は、0 を返します。

#### 説明

画像入力後で、画像変換補正をします。  
但し、一部のカメラのみ必要となります。

\*\*\*\*\*

44) Mvc12\_IsNeedConvert  
    BOOL Mvc12\_IsNeedConvert (void)

引数

なし

戻り値

正常終了の時は、TRUE を返します。

説明

画像入力後で、画像変換補正が必要かどうかをチェックします。  
但し、一部のカメラのみ必要となります。

\*\*\*\*\*

45) Mvc12\_SetScanRate

int Mvc12\_SetScanRate (int data)

引数

data      スキャンレート

戻り値

以前の設定値を返します。

説明

ScanRate の値を設定します。  
次の画像入力から影響します。

\*\*\*\*\*

46) Mvc12\_GetScanRate  
int Mvc12\_GetScanRate (void)

引数  
なし

戻り値  
現在の設定値を返します。

説明  
ScanRate の値を取得します。

\*\*\*\*\*

47) Mvc12\_SetDspCommand  
int Mvc12\_SetDspCommand (int mode)

引数

mode	DSP 設定値
0:	(DSP / 画像入力) BANK 交互選択
1:	DSP BANK0 選択    画像入力 BANK1 選択
2:	DSP BANK1 選択    画像入力 BANK0 選択
3:	DSP BANK0 選択
4:	DSP BANK1 選択
5:	NOT ACTIVE

戻り値

以前の設定値を返します。

説明

DSP コマンドを設定します。  
次の画像入力から影響します。

\*\*\*\*\*

48) Mvc12\_GetDspCommand  
int Mvc12\_GetDspCommand (void)

引数

なし

戻り値

DSP 設定値

0: (DSP / 画像入力) BANK 交互選択

1: DSP BANK0 選択 画像入力 BANK1 選択

2: DSP BANK1 選択 画像入力 BANK0 選択

3: DSP BANK0 選択

4: DSP BANK1 選択

5: NOT ACTIVE

説明

DSP コマンドを取得します。

\*\*\*\*\*

#### 49) Mvc12\_SetLutMode

int Mvc12\_SetLutMode( int mode )

#### 引数

mode            0 : 変換なし  
                 1 : シェーディング変換有効  
                 2 : LUT変換有効

#### 戻り値

正常終了の時は、0 を返します。それ以外は、以下の値を返します。  
1:パラメータエラー

#### 説明

画像入力時のデータ変換モードを指定します。  
この関数を使用する場合は、Mvc12\_SetShdEnable を使用しないでください

\*\*\*\*\*

50) Mvc12\_GetLutMode

void Mvc12\_GetLutMode( int \*mode )

引数

mode            0 : 変換なし  
                 1 : シェーディング変換有効  
                 2 : LUT変換有効

戻り値

なし

説明

現在の画像入力時のデータ変換モードを取得します。

\*\*\*\*\*

51) Mvc12\_LoadLutData

int Mvc12\_LoadLutData(char \*szFullPath)

引数

szFullPath           LUT用ファイルの絶対パス

戻り値

- 0 : 正常終了
- 1 : パラメータエラー
- 2 : ファイルオープンエラー
- 3 : ファイル形式が違います
- 4 : 内部メモリ確保エラー
- 5 : 初期化エラー

説明

LUTデータファイルを読み込み、  
MV-12メモリに格納します。

\*\*\*\*\*

## 52) Mvc12\_SaveLutData

int Mvc12\_SaveLutData(char \*szFullPath)

### 引数

szFullPath            LUT用データファイルの絶対パス

### 戻り値

- 0 : 正常終了
- 1 : パラメータエラー
- 2 : ファイルオープンエラー
- 3 : ファイル形式が違います
- 4 : 内部メモリ確保エラー
- 5 : 初期化エラー

### 説明

現在MV-12メモリに格納されているLUTデータを、  
LUTデータファイルにして保存します。

\*\*\*\*\*

### 53) Mvc12\_SetLutData

int Mvc12\_SetLutData(DWORD\* pSrc, int length)

#### 引数

pSrc            L U T データ配列  
length          L U T データの配列数  
                 最大値は、4 0 9 6 です。  
                 カメラの入力データ長が 8 b i t の時は、2 5 6  
                 カメラの入力データ長が 1 0 b i t の時は、1 0 2 4  
                 カメラの入力データ長が 1 2 b i t の時は、4 0 9 6  
                 を指定します。

#### 戻り値

0 : 正常終了  
1 : 検索エラー  
2 : データ設定値エラー

#### 説明

L U T データを設定します。

\*\*\*\*\*

54) Mvc12\_GetLutData

int Mvc12\_GetLutData(DWORD\* pDst, int length)

引数

pSrc L U T データ配列

length L U T データ配列数

最大値は、4 0 9 6 です。

カメラの入力データ長が 8 b i t の時は、2 5 6

カメラの入力データ長が 1 0 b i t の時は、1 0 2 4

カメラの入力データ長が 1 2 b i t の時は、4 0 9 6  
を指定します。

戻り値

なし

説明

現在のMV-12へ格納されているLUTデータを取得します。

\*\*\*\*\*

55) Mvc12\_GetHist

int Mvc12\_GetHist(int iFrm, RECT\* lprc, unsigned int \*hdata)

引数

iFrm	画像フレーム番号(0 ~ dwFrameNum-1)
lprc	ヒストグラム処理範囲
hdata	ヒストデータ... 配列サイズは、4096以上にしてください

戻り値

- 0 : 正常終了
- 1 : 検索エラー
- 2 : 画像アクセスエラー

説明

ヒストグラムデータを取り込みます。

\*\*\*\*\*

56) Mvc12\_GetHistValidArea

```
int Mvc12_GetHistValidArea(unsigned int *hdata, int SrcX, int SrcY, int *DstX, int *DstY,
                           int mode, double dOff)
```

引数

hdata                    ヒストデータ... 配列サイズは、4096以上にしてください  
 SrcS, SrcE              ヒストデータ開始,終了位置  
                          この輝度階調範囲をを計算対象とします。

\*DstS, \*DstE 求めたデータ有効領域を格納します。

mode                    0  計算パターン0  オフセット式  
                          off以上のデータ数がある範囲を有効としよう。  
                          1  未使用  
                          後で修正予定 標準偏差の  $\pm \sigma * off$  の範囲を有効とします。

off

戻り値

0 : 正常終了  
 1 : パラメータエラー

説明

ヒストグラムデータから、データ有効領域を求めます。  
 ゲイン調整 (LUT変換) の補助関数です

\*\*\*\*\*

### 57) Mvc12\_GetAcquireStopEvent

```
int Mvc12_GetAcquireStopEvent(HANDLE* pObject);
```

#### 引数

\*pObject      同期イベント

#### 戻り値

0 固定

#### 説明

画像入力中止状態を取得するための同期イベントを取得します。

このイベントにより、画像入力のし直しするためのものです。

このイベントは、ライブラリで管理しているため、ユーザーでハンドルを解放する必要はありません。

\*\*\*\*\*

58) Mvc12\_GetDmaPosition

int Mvc12\_GetDmaPosition ( MV12\_DMA\_POS \*pPos )

引数

\* pPos           現在のDMA転送位置 (画像フレーム番号、X座標、Y座標)

戻り値

0 : 正常終了  
1 : 検索エラー

説明

現在のDMA転送位置(DMA転送中のメモリアドレス)を取得できます。  
画像入力実行中に使用します。  
Y座標は、3程度大きい値になることがあります。

\*\*\*\*\*

59) Mvc12\_GetImgValidStatus  
void Mvc12\_GetImgValidStatus ( BOOL \*pDat )

引数

\*pDat            画像入力状態 (0 : 終了   1 : 画像入力中)

戻り値

なし

説明

現在の画像入力状態を取得できます。

※画像入力中は、画像サイズの変更を行わないでください。

例)

```
Mvc12_Acquire(...)            ... 画像入力開始
.....
.....
.....
Mvc12_AcquireCancel()        ... 画像入力停止

BOOL bBusy=TRUE;
while( bBusy )
    Mvc12_GetImgValidStatus( &bBusy );

Mvc12_SetCameraConfig2() ... 画像サイズ変更

Mvc12_Acquire(...)            ... 画像入力開始
```

# 株式会社マイクロビジョン

〒950-0986 新潟市神道寺南1-9-15

TEL 025-247-8312

FAX 025-247-8313