

Mileva シリーズ
Software Developer Kit
MX-02

株式会社クリオ

初版 2017年8月1日
第2版 2018年6月8日
第3版 2018年7月17日
第4版 2018年9月18日
第5版評価版 2020年9月14日
第6版評価版 2023年7月28日
第7版評価版 2026年6月15日

来歴

- '17.8.1 初版(DLL Version 1.0.0.0)
- '18.6.8 第2版(DLL Version 1.0.0.0)
 - 64bit 版(DLL Version 1.0.0.1)追加
- '18.7.17 第3版(DLL Version 1.0.1.0)
 - 複数台対応のため、下記の関数を追加
 - MtuOpenSerialNo 関数追加
 - MtuGetDeviceNameWithSerialNo 関数追加
- '18.9.18 第4版(DLL Version 1.0.1.1)
 - ソフトウェア修正内容
 - MtuSetCameraNo 関数にて、1カメラ仕様にて No.2 が設定できる不具合修正
 - MtuSetExposure 関数にて 設定値 MAX が"1"になっていたのを"0"に修正
 - 本仕様書誤記訂正
 - MtuOpenSerialNo 関数 *type 記述の消去
 - MtuSetExposure 関数 露出時間設定範囲 MAX "0"へ変更
- '20.9.14 第5版評価版(DLL Version 1.1.0.0)
 - Mileva2 Sensor(MS-111) (MS-211)対応
 - MtuOpenSerialNo 変更
 - シェーディング補正機能、画素欠陥補正機能追加(Mileva2 Sensor のみ)
 - MtuGetPixelDefectOnOff, MtuSetPixelDefectOnOff 関数追加**
 - MtuGetShadingCorrectionOnOff, MtuSetShadingCorrectionOnOff 関数追加**
- '23.7.28 第6版評価版(DLL Version 1.1.1.0)
 - 多項式フィッティング機能追加
 - MtuGetPolynomialFittingOnOff, MtuSetPolynomialFittingOnOff 関数追加**
- '26.6.15 第7版評価版(DLL Version 1.2.0.2)
 - Mileva5 追加
 - MtuXGetDigitalGain, MtuXSetDigitalGain 関数追加
 - MtuXGetDigitalGainMax, MtuXGetDigitalGainMin 関数追加
 - MtuGetShutterTime, MtuSetShutterTime 関数追加
 - MtuGetShutterMax, MtuGetShutterMin 関数追加
 - Mtu5GetAnalogGain, Mtu5SetAnalogGain 関数追加
 - 多項式フィッティング機能、シェーディング補正機能削除

目次

1	概要.....	6
2	対応機器構成.....	7
2.1	対応 OS	7
2.2	対応開発言語.....	7
2.3	ライブラリ構成.....	7
2.4	対応機種.....	8
3	使用方法.....	9
4	関数.....	10
4.1	関数一覧.....	10
4.2	MtuOpen.....	12
4.3	MtuOpenSerialNo.....	13
4.4	MtuClose.....	14
4.5	MtuGetBufferSize.....	14
4.6	MtuGetDevice.....	15
4.7	MtuGetDeviceWithSerialNo	15
4.8	MtuGetFormat.....	16
4.9	MtuReceiveImage.....	17
4.10	MtuSetLed.....	18
4.11	MtuGetLed.....	18
4.12	MtuSetGain.....	19
4.13	MtuGetGain.....	19
4.14	MtuXSetGain.....	20
4.15	MtuXGetGain.....	20
4.16	MtuXGetGainMax.....	21
4.17	MtuXGetGainMin.....	21
4.18	MtuSetExposure.....	22
4.19	MtuGetExposure.....	23
4.20	MtuSetCameraNo.....	24
4.21	MtuGetCameraNo.....	24
4.22	MtuSetColorBar.....	25
4.23	MtuGetColorBar.....	25
4.24	MtuCameraRecognition	26

4.25	MtuUSBMode.....	27
4.26	MtuGPIODirection	28
4.27	MtuGPIOOutput.....	29
4.28	MtuGPIOInput.....	30
4.29	MtuTemperature.....	31
4.30	MtuSetDigitalGain	32
4.31	MtuGetDigitalGain	33
4.32	MtuXSetDigitalGain	34
4.33	MtuXGetDigitalGain	35
4.34	MtuXGetDigitalGainMax	36
4.35	MtuXGetDigitalGainMin	37
4.36	MtuSetShutter.....	38
4.37	MtuGetShutter.....	39
4.38	MtuGetShutterMax	40
4.39	MtuGetShutterMin	40
4.40	MtuSetShutterTime	41
4.41	MtuGetShutterTime	42
4.42	MtuGetSerialNo.....	42
4.43	MtuSetPixelDefectOnOff	43
4.44	MtuGetPixelDefectOnOff	43
4.45	Mtu5SetAnalogGain	44
4.46	Mtu5GetAnalogGain	44
4.47	MtuVersion.....	45
4.48	MtuDLLVersion.....	45
5	取得画像について.....	46
5.1	RAW データ構成	46
5.2	画素構成.....	46
6	使用方法.....	47
6.1	Visual C++ 静的リンク	47
6.1.1	プロジェクトの作成	47
6.1.2	ヘッダファイルのインクルード	47
6.1.3	ライブラリファイルのインクルード	47
6.2	C#.....	48

6.2.1	プロジェクトの作成	48
6.2.2	ファイルの登録.....	48
6.3	OpenCV との提携方法	49
6.4	Python サンプル.....	50
7	注意事項.....	51
7.1	色空間タイプ（メディアサブタイプ）	51
7.2	他の UVC カメラの動作.....	51
8	Mileva シリーズの相違点	51

本書に関する注意事項

1. ご使用の前に必ず本書をよくお読みになり、注意事項を確認のうえ製品を正しくご使用ください。
2. 本書は必要なときに参照できるよう、大切に保管してください。
3. 本製品を本来の目的以外の用途でお使いになった場合には、安全性を保証致しかねますので、ご了承ください。
4. 本書の内容に関して、将来予告なしに変更することがあります。
5. 本書に掲載している図は、説明のため、一部、省略や抽象化を行っています。
6. 本書の内容に関して、万一ご不審な点や誤り、記載漏れなどお気づきのことがございましたら、本書末尾記載の弊社連絡先または販売店までご連絡ください。
7. 本書の内容の一部または全部を、弊社に無断で転載・複製・改変することはできません。

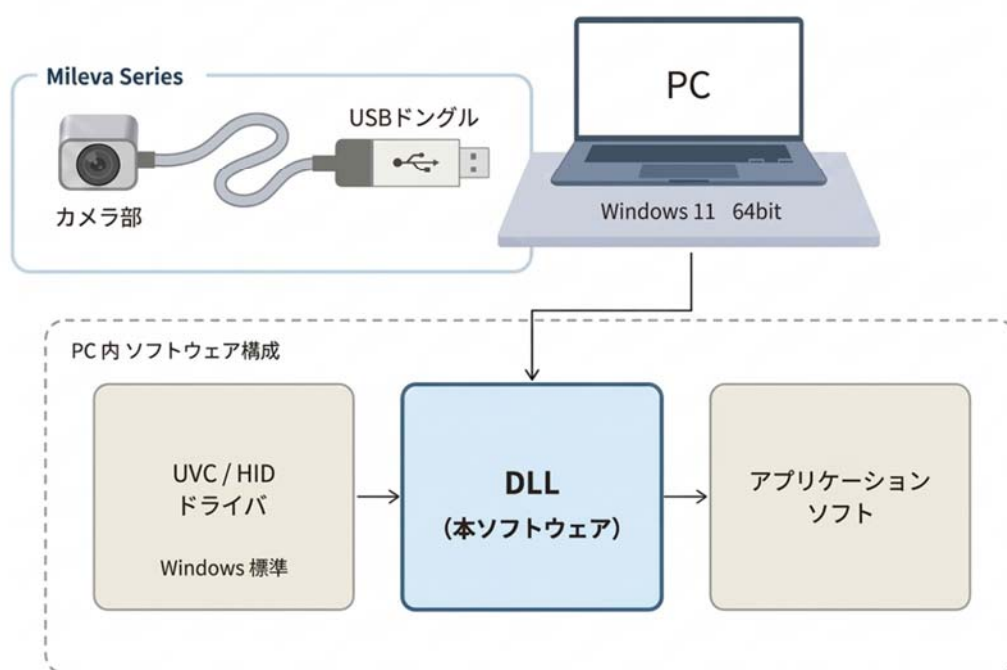
弊社では、本製品を運用した結果での損失、逸失利益等の請求につきましては、項目 6 に関わらずいかなる責任も負いかねますので、予めご了承ください。

1 概要

本書は、Mileva シリーズを使用するためのソフトウェアデベロッパーキット (SDK) についての説明をします。

本 SDK は、ダイナミックリンクライブラリ (DLL) 形式での提供になります。

本 DLL は、ドライバーアプリケーションソフト間を制御しやすくするためのものです。



2 対応機器構成

2.1 対応 OS

Windows11 64bit

2.2 対応開発言語

C++	Visual Studio Community 2022 にて動作確認済み
C#.NET	Visual Studio Community 2022 にて動作確認済み
Python	Version 3.14.4 にて動作確認済み

2.3 ライブラリ構成

MilevaSensor SDK.pdf 本書

x86 フォルダ(32bit バージョン)

mtu.dll	DLL 本体
mtu.h	C++用ヘッダーファイル
mtu.lib	C++用ライブラリ
mtu.cs	C#用クラス
mtu.py	Python 用 import ファイル

x64 フォルダ(64bit バージョン)

mtu.dll	DLL 本体
mtu.h	C++用ヘッダーファイル
mtu.lib	C++用ライブラリ
mtu.cs	C#用クラス
mtu.py	Python 用 import ファイル

各 DLL は、Windows の system ディレクトリもしくは、使用アプリケーションの同一ディレクトリにコピーして使用してください。

2.4 対応機種

本 DLL は下記の機種に対応しています。

Mileva Sensor	MS-011	1 カメラモデル (カメラ形状	うちわ型)
	MS-012	2 カメラモデル (カメラ形状	うちわ型)
Mileva2 Sensor	MS-211	1 カメラモデル (カメラ形状	円筒型)
Mileva3 Sensor	MS-311	1 カメラモデル (カメラ形状	角型)
Mileva4 Sensor	MS-411	1 カメラモデル (カメラ形状	長方形)
Mileva5 Sensor	MS-511	1 カメラモデル (カメラ形状	角型)

Mileva シリーズは、上記の全モデルを示します。

※各モデルの詳細な仕様については、各モデルの仕様書をご確認ください。

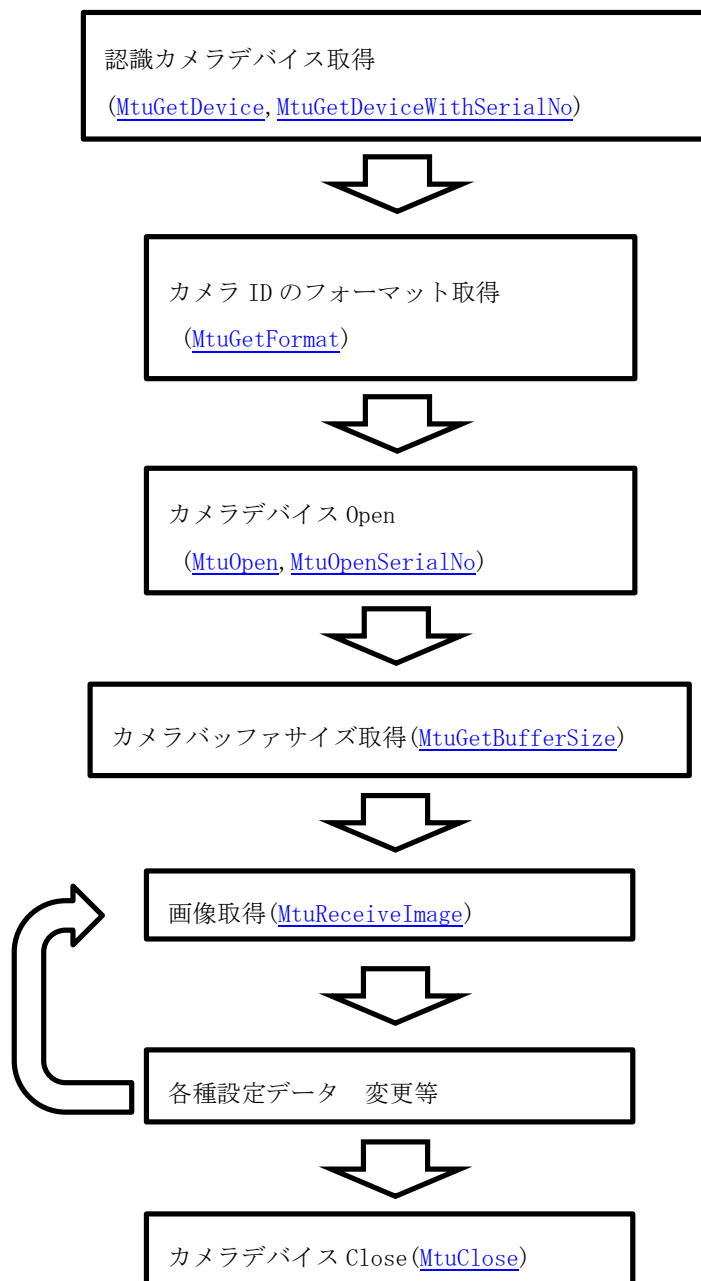
※以下、特に指定なきものについては

Mileva Sensor Mileva
Mileva2 Sensor Mileva2
Mileva3 Sensor Mileva3
Mileva4 Sensor Mileva4
Mileva5 Sensor Mileva5

として表記します。

3 使用方法

DLL を使用して画像を取得するためのフローを記述します。



4 関数

下記に DLL 内にある関数群の説明を記述します。

関数については、C++を基準として記述します。

4.1 関数一覧

関数名	説明	対応機種
MtuOpen	オープン	全て
MtuOpenSerialNo	オープン(シリアル No.)	全て
MtuClose	クローズ	全て
MtuGetBufferSize	バッファサイズ取得	全て
MtuGetDevice	カメラ認識デバイス取得	全て
MtuGetDeviceWithSerialNo	カメラ認識デバイス取得(シリアルNo. 付)	全て
MtuGetFormat	カメラフォーマット取得	全て
MtuReceiveImage	画像データ受信	全て
MtuSetLed	照明設定	全て
MtuGetLed	照明取得	全て
MtuSetGain	ゲイン設定	全て
MtuGetGain	ゲイン取得	全て
MtuXSetGain	ゲイン詳細 設定	全て
MtuXGetGain	ゲイン詳細 取得	全て
MtuXGetGainMax	ゲイン詳細 MAX 値取得	全て
MtuXGetGainMin	ゲイン詳細 MIN 値取得	全て
MtuSetExposure	露出設定	全て
MtuGetExposure	露出取得	全て
MtuSetCameraNo	カメラ No. 設定	Mileva のみ
MtuGetCameraNo	カメラ No. 取得	全て
MtuSetColorBar	カラーバー設定	全て
MtuGetColorBar	カラーバー取得	全て
MtuCameraRecognition	カメラ認識取得	全て
MtuUSBMode	USB モード取得	全て
MtuGPIODirection	GPIO 入出力設定	Mileva のみ
MtuGPIOOutput	GPIO 出力設定	Mileva のみ

関数名	説明	対応機種
MtuGPIOInput	GPIO 入力	Mileva のみ
MtuTemperature	温度取得	Mileva/2/3/5 のみ
MtuSetDigitalGain	digital_gain の設定	全て
MtuGetDigitalGain	digital_gain の読込	全て
MtuXSetDigitalGain	digital_gain 詳細設定	全て
MtuXGetDigitalGain	digital_gain 詳細読込	全て
MtuXSetDigitalGainMax	digital_gain 詳細 MAX 値取得	全て
MtuXGetDigitalGainMin	digital_gain 詳細 MIN 値取得	全て
MtuSetShutter	電子シャッター設定	全て
MtuGetShutter	電子シャッター読込	全て
MtuGetShutterMax	電子シャッター設定 Max 値取得	全て
MtuGetShutterMin	電子シャッター読込 Min 値取得	全て
MtuSetShutterTime	電子シャッター時間単位設定	全て
MtuGetShutterTime	電子シャッター時間単位読込	全て
MtuGetSerialNo	シリアル No 読込	全て
MtuSetPixelDefectOnOff	画素欠陥 ON/OFF 設定	Mileva2/3/4/5 のみ
MtuGetPixelDefectOnOff	画素欠陥 ON/OFF 読込	Mileva2/3/4/5 のみ
Mtu5SetAnalogGain	アナログゲインの設定	Mileva5 のみ
Mtu5GetAnalogGain	アナログゲインの取得	Mileva5 のみ
MtuVersion	バージョン情報読込	全て
MtuDLLVersion	DLL バージョン情報読込	全て

4.2 MtuOpen

カメラをオープンします。

定義： int MtuOpen(int CameraId ,int width ,int height ,double fps,char *type)

引数： CamraId カメラ ID
 width 画像 幅
 height 画像 高さ
 fps フレームレート
 *type 色空間タイプ (画像フォーマット)

戻り値： -1:異常 0:Mileva シリーズ 1: Mileva シリーズ以外

詳細： 認識しているカメラ ID を指定する必要があります。

 (認識しているカメラは、MtuGetDevice にて取得可能です)

カメラの画像の幅・高さ・フレームレートを指定します。

本カメラは

Mileva/2/3/4

1280x720	USB3.0	60/30fps	USB2.0	10/5fps
656x408	USB3.0	120/60fps	USB2.0	30/15fps
328x204	USB3.0	240/120fps	USB2.0	60/40fps

Mileva5

1920x1080	USB3.0	30/15fps	USB2.0	10/5fps
960x540	USB3.0	90/45fps	USB2.0	30/15fps
480x270	USB3.0	180/90fps	USB2.0	60/30fps

が指定可能です。

色空間タイプは 文字列 “YUY2”を指定します。

※カメラをオープンできる最大数は10台までです。

4.3 MtuOpenSerialNo

カメラのシリアル No. を使用してオープンします。

定義：`int MtuOpenSerialNo(char *serialno,int *CameraId ,int width ,int height ,double fps)`

引数:	serialno	シリアル No.
	*CamraId	カメラ ID
	width	画像 幅
	height	画像 高さ
	fps	フレームレート

戻り値: -1:異常 0:正常

詳細： カメラのシリアル No. を指定する必要があります。
機種コード+シリアル No. 8 桁の文字列です。
機種コードは “MTU#”, ”MTU2#”, ”MTU5#” の文字列になります。
カメラ ID は返却値となります。
カメラの画像の幅・高さ・フレームレートを指定します。
本カメラは

Mileva/2/3/4

1280x720	USB3.0	60/30fps	USB2.0	10/5fps
656x408	USB3.0	120/60fps	USB2.0	30/15fps
328x204	USB3.0	240/120fps	USB2.0	60/40fps

Milleva5

1920x1080	USB3.0	30/15fps	USB2.0	10/5fps
960x540	USB3.0	90/45fps	USB2.0	30/15fps
480x270	USB3.0	180/90fps	USB2.0	60/30fps

が指定可能です。

※Mileva シリーズのみオープン可能です。

※カメラをオープンできる最大数は10台までです。

※Ver 1.2.0.0 以降、機種コード

Mileva	“MTU#”
Mileva2/3/4	”MTU2#”
Mileva5	“MTU5#”

を指定する必要があります。

4.4 MtuClose

カメラをクローズします。

定義： void MtuClose(int CameraId)
引数： CameraId カメラ ID
戻り値： なし

詳細： MtuOpen にてオープンされたカメラデバイスをクローズします。

4.5 MtuGetBufferSize

カメラの受信画像のバッファサイズを取得します。

定義： int MtuGetBufferSize(int CameraId)
引数： CameraId カメラ ID
戻り値： 0 : エラー 0 以外 : 正常

詳細： MtuOpen にてオープンされたカメラの画像データの受信サイズをバイト単位にて
取得できます。
MtuOpen にてオープンされていない場合は、0 が返ります。

Mileva シリーズは下記の画像サイズ対応のため

Mileva/2/3/4

1280x720x2	⇒	1,843,200 byte
656x408x2	⇒	535,296 byte
328x204x2	⇒	133,824 byte

Mileva5

1920x1080x2	⇒	4,147,200 byte
960x540x2	⇒	1,036,800 byte
480x270x2	⇒	259,200 byte

が返ってきます。

4.6 MtuGetDevice

認識しているカメラデバイスを取得します。

定義: int MtuGetDevice(char *devicename ,int size)

引数: *devicename 取得デバイス名

 size 取得デバイス名 文字列の最大容量

戻り値: 認識数量 0=無 -1:エラー

詳細: 認識しているカメラデバイスのデバイス名をカンマ(“,”)区切りにて文字列で取得します。

 この認識順序がカメラ ID(CameraId)となります。

 カメラ ID の数値は、0 始まりです。

 取得デバイス名が文字列の最大容量以上になった場合は、エラー(-1)になります。

4.7 MtuGetDeviceWithSerialNo

認識しているカメラデバイスを取得します。Mileva シリーズにシリアル No が付属されます。

定義: int MtuGetDeviceWithSerialNo(char *devicename ,int size)

引数: *devicename 取得デバイス名

 size 取得デバイス名 文字列の最大容量

戻り値: 認識数量 0=無 -1:エラー

詳細: 認識しているカメラデバイスのデバイス名をカンマ(“,”)区切りにて文字列で取得します。

 この認識順序がカメラ ID(CameraId)となります。

 カメラ ID の数値は、0 始まりです。

 取得デバイス名が文字列の最大容量以上になった場合は、エラー(-1)になります。

Mileva "MTU#xxxxxxx"(x=シリアル No 数値 8 桁)

Mileva2/3/4 "MTU2#xxxxxxx"(x=シリアル No 数値 8 桁)

Mileva5 "MTU5#xxxxxxx"(x=シリアル No 数値 8 桁)

4.8 MtuGetFormat

カメラデバイス内の使用できるフォーマットを文字列にして取得します。

定義: `bool MtuGetFormat(int CameraId, char *format, int size)`

引数: CameraId カメラ ID

 *format フォーマット

 size フォーマット 文字列の最大容量

戻り値: true:正常 false:異常

詳細: カメラ ID にて設定したカメラにて使用できるフォーマットを文字列にて取得します。

 文字列は、カンマ(“,”)区切りです。

 文字列順序

 No. , 高さ , 幅 , フレームレート , 色空間タイプ ,

 <例>

 No. =0, 高さ=720, 幅=1280, フレームレート=60, 色空間タイプ=YUY2

 0, 720, 1280, 60.0000, YUY2,

 取得デバイス名が文字列の最大容量以上になった場合は、false になります。

4.9 MtuReceiveImage

画像データを取得します。

定義: `bool MtuReceiveImage (int CameraId , unsigned char *data)`

引数: CameraId カメラ ID
*data 画像データ

戻り値: true:正常 false:異常 (未取得含む)

詳細: カメラ ID における画像データを取得します。
画像データは、MtuGetBufferSize にて受信した容量よりも多くのエリアを
確保する必要があります。

画像が取得できない場合またはフレームレート間隔以内に取得しようとした場合
戻り値には false が返ります。
取得した画像データの配列については、5 取得画像について を参照してください。

※Mileva2/3/4/5 では、画素欠陥補正が追加 (ON) になった画像がデフォルトで
取得されます。

4.10 MtuSetLed

照明用の LED の明るさを設定します。

定義： `bool MtuSetLed(int CameraId ,int level)`
引数： `CameraId` カメラ ID
 `level` 明るさ 0 ～ 6 4 (default:0)
戻り値： `true`:正常 `false` : 異常

詳細： カメラ ID における照明 (LED) の明るさを調整します。
 0 (消灯) 1 (暗) ～ 6 4 (明)

※Mileva 2 カメラモデル (MS-012) で両カメラ、点灯します
※MtuOpen にてオープンしてある必要があります。

4.11 MtuGetLed

照明用の LED の明るさを取得します。

定義： `bool MtuGetLed(int CameraId ,int *level)`
引数： `CameraId` カメラ ID
 `*level` 明るさ 0 ～ 6 4 (default:0)
戻り値： `true`:正常 `false` : 異常

詳細： 現在のカメラ ID における照明 (LED) の明るさを取得します。
 0 (消灯) 1 (暗) ～ 6 4 (明)

※Mileva 2 カメラモデル (MS-012) で両カメラ、点灯します
※MtuOpen にてオープンしてある必要があります。

4.12 MtuSetGain

カメラのゲインを設定します。

定義: `bool MtuSetGain(int CameraId ,int level)`

引数: `CameraId` カメラ ID
 `level` ゲイン 0 ～ 2 4 0

戻り値: `true`:正常 `false`: 異常

詳細: カメラ ID におけるカメラのゲインを調整します。
 0 (低) ～ 2 4 0 (高)

※カメラ認識数が2つの場合、現在、設定中のカメラを設定します。

※MtuOpen にてオープンしてある必要があります。

※Mileva5 は、実際には 0-4096 まで設定できるが、0-960 までの 4step 単位としています。

詳細な設定が必要な場合、[MtuXSetGain](#) を使用してください。

4.13 MtuGetGain

カメラのゲインを取得します。

定義: `bool MtuGetGain(int CameraId ,int *level)`

引数: `CameraId` カメラ ID
 `*level` ゲイン 0 ～ 2 4 0

戻り値: `true`:正常 `false`: 異常

詳細: 現在のカメラ ID におけるカメラのゲイン値を取得します。
 0 (低) ～ 2 4 0 (高)

※カメラ認識数が2つの場合、現在、設定中のカメラの値を取得します。

※MtuOpen にてオープンしてある必要があります。

※Mileva5 は、0-4096 まで設定できるが、0-960 までの 4step 単位(1/4 の値)としています。

詳細な設定が必要な場合、[MtuXGetGain](#) を使用してください。

[MtuXSetGain](#) にて詳細値を設定した値は、4 で割った値及び 960 以上は 960 を返します

4.14 MtuXSetGain

カメラのゲインを詳細値にて設定します。

定義: `bool MtuXSetGain(int CameraId ,int level)`

引数: `CameraId` カメラ ID
 `level` ゲイン

戻り値: `true`:正常 `false`: 異常

詳細: カメラ ID におけるカメラのゲインの詳細値を調整します。

Mileva/2/3/4 0 (低) ～ 2 4 0 (高)
Mielva5 0 (低) ～ 4 0 9 5 (高)

※カメラ認識数が2つの場合、現在、設定中のカメラを設定します。

※MtuOpen にてオープンしてある必要があります。

4.15 MtuXGetGain

カメラのゲインを詳細値にて取得します。

定義: `bool MtuXGetGain(int CameraId ,int *level)`

引数: `CameraId` カメラ ID
 `*level` ゲイン

戻り値: `true`:正常 `false`: 異常

詳細: 現在のカメラ ID におけるカメラのゲインの詳細値を取得します。

Mileva/2/3/4 0 (低) ～ 2 4 0 (高)
Mielva5 0 (低) ～ 4 0 9 5 (高)

※カメラ認識数が2つの場合、現在、設定中のカメラの値を取得します。

※MtuOpen にてオープンしてある必要があります。

4.16 MtuXGetGainMax

カメラのゲイン詳細値の最大(MAX)値を取得します。

定義: `bool MtuXGetGainMax(int CameraId ,int *level)`

引数: CameraId カメラ ID
*level ゲイン最大(MAX) 値

戻り値: true:正常 false: 異常

詳細: 現在のカメラ ID におけるカメラのゲインの最大(MAX) 値を取得します。

Mileva/2/3/4	2 4 0
Mileva5	4 0 9 5

※MtuOpen にてオープンしてある必要があります。

4.17 MtuXGetGainMin

カメラのゲイン詳細値の最小(MIN)値を取得します。

定義: `bool MtuXGetGainMin(int CameraId ,int *level)`

引数: CameraId カメラ ID
*level ゲイン最小(MIN) 値

戻り値: true:正常 false: 異常

詳細: 現在のカメラ ID におけるカメラのゲイン値を取得します。

Mileva/2/3/4	0
Mileva5	0

※MtuOpen にてオープンしてある必要があります。

4.18 MtuSetExposure

カメラの露出を設定します。

定義: `bool MtuSetExposure(int CameraId, int level)`

引数: `CameraId` カメラ ID

`level` 露出 - 13 ~ 0 (default: -6)

戻り値: `true`: 正常 `false`: 異常

詳細: カメラ ID におけるカメラの露出(電子シャッター)を調整します。

 - 13 (低) ~ 0 (高)

 露出時間は下記になります。

0	2^0	1sec
-1	$1/2^1$	500msec
-2	$1/2^2$	250msec
-3	$1/2^3$	125msec
-4	$1/2^4$	62.5msec
-5	$1/2^5$	31.25msec
-6	$1/2^6$	15.63msec
-7	$1/2^7$	7.813msec
-8	$1/2^8$	3.906msec
-9	$1/2^9$	1.953msec
-10	$1/2^{10}$	976.6usec
-11	$1/2^{11}$	488.3usec
-12	$1/2^{12}$	244.1usec
-13	$1/2^{13}$	122.1usec

※カメラ認識数が2つの場合、現在、設定中のカメラを設定します。

※MtuOpen にてオープンしてある必要があります。

※カメラのフレームレートと露出時間の組み合わせにより、フレームレートが遅くなります。

＜例＞フレームレートが 60fps で 露出時間を -3 に設定した場合

※露出時間が有効でない場合があります。(Mileva/2/3/4)

1280x720 5fps 露出時間 -13 -12 の値を設定

4.19 MtuGetExposure

カメラの露出値を取得します。

定義: `bool MtuGetExposure(int CameraId ,int *level)`
引数: `CameraId` カメラ ID
 `*level` 露出 - 1 3 ~ 0 (default:- 6)
戻り値: `true`:正常 `false`: 異常

詳細: 現在のカメラ ID におけるカメラの露出の値を取得します。
 - 1 3 (低) ~ 0 (高)
 数値の詳細については、MtuSetExposure をご確認ください。

※カメラ認識数が2つの場合、現在、設定中のカメラの状態を取得します。

※MtuOpen にてオープンしてある必要があります。

4.20 MtuSetCameraNo

カメラの切り替えを行います。

定義: `bool MtuSetCameraNo(int CameraId, int no)`

引数: `CameraId` カメラ ID
 `no` カメラ No. 1・2

戻り値: `true`:正常 `false`:異常

詳細: カメラを1もしくは2に切り替えます。
 カメラが1つの場合、1もしくは2に固定されます。そのため、接続されていない
 カメラ側を設定すると異常となります。

※MtuOpen にてオープンしてある必要があります。

4.21 MtuGetCameraNo

現在のカメラ No. を取得します。

定義: `bool MtuGetCameraNo(int CameraId, int *no)`

引数: `CameraId` カメラ ID
 `*no` カメラ No. 1・2

戻り値: `true`:正常 `false`:異常

詳細: 現在、設定されているカメラの No. を取得します。

※MtuOpen にてオープンしてある必要があります。

4.22 MtuSetColorBar

カメラの画像をカラーバーにします。

定義: `bool MtuSetColorBar(int CameraId, int onoff)`

引数: `CameraId` カメラ ID
 `onoff` 0 : OFF 1 : ON

戻り値: `true`:正常 `false`:異常

詳細: 1 (ON)を設定するとカメラからの画像がカラーバーになります。

※カメラ認識数が2つの場合、現在、設定中のカメラを設定します。

※MtuOpen にてオープンしてある必要があります。

※Mileva5 はパターン表示となります。

4.23 MtuGetColorBar

現在のカメラからの画像が、カラーバーの状態を取得します。

定義: `bool MtuGetColorBar(int CameraId, int *onoff)`

引数: `CameraId` カメラ ID
 `*onoff` 0 : OFF 1 : ON

戻り値: `true`:正常 `false`:異常

詳細: 現在、設定されているカラーバーの状態を取得します。

 1 (ON)の場合、カラーバー状態です。

※カメラ認識数が2つの場合、現在、設定中のカメラの状態を取得します。

※MtuOpen にてオープンしてある必要があります。

※Mileva5 はパターン表示となります。

4.24 MtuCameraRecognition

現在のカメラの個数を取得します。

定義： `bool MtuCameraRecognition(int CameraId ,BYTE *Camera)`

引数： `CameraId` カメラ ID
 `*Camera` カメラ認識

戻り値： `true`:正常 `false`:異常

詳細： 現在、設定されているカメラを取得します。

取得データはそれぞれ、ビット配置にて

`0bit` カメラ No.1

`1bit` カメラ No.2

となっており

`0` : なし `1` : あり

です。

数値としては

`0` : 両方なし

`1` : カメラ1のみ

`2` : カメラ2のみ

`3` : カメラ1・2あり

となります。

※MtuOpen にてオープンしてある必要があります。

4.25 MtuUSBMode

現在、接続されているUSBのモードを取得します。

定義: `bool MtuUSBMode(int CameraId, BYTE *usb)`

引数: `CameraId` カメラ ID

`*usb` USBモード

戻り値: `true`:正常 `false`:異常

詳細: 現在、接続されているUSBのモードを取得します。

 下記は取得した値による各モード表記になります。

- | | | |
|---|---------------------|----------------------|
| 1 | FULL SPEED(USB2.0) | ※機器が動作しないため、取得されません。 |
| 2 | HIGH SPEED(USB2.0) | |
| 3 | SUPER SPEED(USB3.0) | |

※MtuOpen にてオープンしてある必要があります。

4.26 MtuGPIODirection

外部接続 G P I O ポートの入出力を設定します。

定義： `bool MtuGPIODirection(int CameraId ,BYTE Dir)`

引数： `CameraId` カメラ ID

`Dir` 入出力 各ビット 0 ～ 7 迄 0 : 入力 1 : 出力

戻り値： `true`:正常 `false` : 異常

詳細： 外部接続の G P I O ポートの入出力を設定します。

GPI00 0 bit 0 : 入力 1 : 出力

GPI01 1 bit 0 : 入力 1 : 出力

GPI02 2 bit 0 : 入力 1 : 出力

GPI03 3 bit 0 : 入力 1 : 出力

GPI04 4 bit 0 : 入力 1 : 出力

GPI05 5 bit 0 : 入力 1 : 出力

GPI06 6 bit 0 : 入力 1 : 出力

GPI07 7 bit 0 : 入力 1 : 出力

<例>

GPI01, 2, 4, 6, 7 が入力、GP00, 3, 5 が出力の時の数値は 00101001b=0x29

※デフォルトの設定はすべて入力になっています。

※MtuOpen にてオープンしてある必要があります。

※Mileva のみ対応。その他の機種では `false`:異常になります。

4.27 MtuGPIOOutput

外部接続G P I Oポートの出力を設定します。

定義： `bool MtuGPIOOutput(int CameraId ,BYTE Output)`

引数： `CameraId` カメラ ID

`Output` 出力 各ビット0～7迄 0 : Lo 1 : Hi

戻り値： `true`:正常 `false` : 異常

詳細： 外部接続のG P I Oポートの出力を設定します。

GPI00 0 bit 0 : Lo 1 : Hi

GPI01 1 bit 0 : Lo 1 : Hi

GPI02 2 bit 0 : Lo 1 : Hi

GPI03 3 bit 0 : Lo 1 : Hi

GPI04 4 bit 0 : Lo 1 : Hi

GPI05 5 bit 0 : Lo 1 : Hi

GPI06 6 bit 0 : Lo 1 : Hi

GPI07 7 bit 0 : Lo 1 : Hi

<例>

GPI01, 2, 4 が Hi、GP00, 3, 5, 6, 7 が Lo の時の数値は 00010110b=0x16

※MtuOpen にてオープンしてある必要があります。

※Mileva のみ対応。その他の機種では `false`:異常になります。

4.28 MtuGPIOInput

外部接続 G P I O ポートの入力値を取得します。

定義： `bool MtuGPIOInput(int CameraId ,BYTE *Input)`

引数： `CameraId` カメラ ID

`Input` 入力 各ビット 0 ～ 7 迄 0 : Lo 1 : Hi

戻り値： `true`:正常 `false` : 異常

詳細： 外部接続の G P I O ポートの入力値を取得します。

GPI00 0 bit 0 : Lo 1 : Hi

GPI01 1 bit 0 : Lo 1 : Hi

GPI02 2 bit 0 : Lo 1 : Hi

GPI03 3 bit 0 : Lo 1 : Hi

GPI04 4 bit 0 : Lo 1 : Hi

GPI05 5 bit 0 : Lo 1 : Hi

GPI06 6 bit 0 : Lo 1 : Hi

GPI07 7 bit 0 : Lo 1 : Hi

<例>

GPI01, 2, 4 が Hi、GP00, 3, 5, 6, 7 が Lo の時の数値は 00010110b=0x16

※出力が設定されたビットの入力値は不定となります。

※MtuOpen にてオープンしてある必要があります。

※Mileva のみ対応。その他の機種では `false`:異常になります。

4.29 MtuTemperature

カメラに付属する温度センサの値を取得します。

定義： `bool MtuTemperature(int CameraId ,int CameraNo ,int *Temp)`

引数： `CameraId` カメラ ID

`CameraNo` 取得するカメラの No. 1 または 2

`*Temp` 温度(℃)

戻り値： `true`:正常 `false`:異常

詳細： カメラに付属する温度センサから温度を取得します。

 温度の単位は 摂氏(℃)です。

 温度は、 -5.5°C ～ $+12.8^{\circ}\text{C}$ 迄の値です。

※MtuOpen にてオープンしてある必要があります。

※Mileva/2/3/5 のみ有効です。その他の機種では `false`:異常になります。

4.30 MtuSetDigitalGain

カメラの digital_gain 値を設定、変更します。

定義: `bool MtuSetDigitalGain (int CameraId ,WORD greenR,WORD red`
`,WORD blue,WORD greenB)`

引数:	CameraId	カメラ ID
	greenR	カメラ画素 Gr の digital_gain 値
	red	カメラ画素 R の digital_gain 値
	blue	カメラ画素 B の digital_gain 値
	greenB	カメラ画素 Gb の digital_gain 値

戻り値: true:正常 false:異常

詳細： カメラの画素の greenR=Gr red=R blue=B greenB=Gb の digital_gain 値を設定します。

digital_gain 値は、

0x100 (256)–0xffff (4095)

の範囲です。

デフォルトの設定値は

	Mileva	Mileva2/3/4	Mileva5
greenR	0x00d4	0x010a	0x640 (4 倍の値)
red	0x0117	0x017a	0xbcc (4 倍の値)
blue	0x0170	0x01ee	0xd54 (4 倍の値)
greenB	0x00f8	0x010e	0x640 (4 倍の値)

となっています。

※現在設定されている CameraNo 側のカメラのみ値が反映されます。

※MtuOpen により値はリセットされます。

※MtuOpen にてオープンしてある必要があります。

※Mileva の設定範囲は 0-65535 (0xffff) ですが、実際の範囲は 0x100 (256)-0xffff (4095) です。

※Mileva5 の実際の設定範囲は 0-1023 (0x3ff) までです。

本関数による、設定値は 0x03f を切り捨てた、4 倍の値を設定します。

設定範囲として 0x03f-0x3ff となります。

詳細な設定を使用する場合 MtuXSetDigitalGain を使用してください。

4.31 MtuGetDigitalGain

カメラの digital_gain 値を読み込みます。

定義: `bool MtuGetDigitalGain (int CameraId ,WORD *greenR,WORD *red`
`,WORD *blue,WORD *greenB)`

引数:	CameraId	カメラ ID
	greenR	カメラ画素 Gr の digital_gain 値
	red	カメラ画素 R の digital_gain 値
	blue	カメラ画素 B の digital_gain 値
	greenB	カメラ画素 Gb の digital_gain 値

戻り値: true:正常 false:異常

詳細： カメラの画素の greenR=Gr red=R blue=B greenB=Gb の digital_gain 値を読み込みます。

digital_gain 値は、

0x100(0x100)-0xffff(4095)

の範囲です。

デフォルトの設定値は

	Mileva	Mileva2/3/4	Mileva5
greenR	0x00d4	0x010a	0x640 (4 倍の値)
red	0x0117	0x017a	0xbcc (4 倍の値)
blue	0x0170	0x01ee	0xd54 (4 倍の値)
greenB	0x00f8	0x010e	0x640 (4 倍の値)

となっています。

※設定されている CameraNo 側のカメラの値です。

※MtuOpen により値はリセットされます。

※MtuOpen にてオープンしてある必要があります。

※Mileva5 の設定範囲は 0-1023 (0x3ff) としています

詳細な値を取得を使用する場合 MtuXGetDigitalGain を使用してください

MtuXGetDigitalGain で設定された値を取得した場合、4 倍の値で LSB 2bit (0x000-0x03f)

は切り捨てとなります

4.32 MtuXSetDigitalGain

カメラの digital_gain 値を詳細値にて設定、変更します。

[illegible]

引数:	CameraId	カメラ ID
	greenR	カメラ画素 Gr の digital_gain 値
	red	カメラ画素 R の digital_gain 値
	blue	カメラ画素 B の digital_gain 値
	greenB	カメラ画素 Gb の digital_gain 値

戻り値: true:正常 false:異常

詳細： カメラの画素の greenR=Gr red=R blue=B greenB=Gb の digital_gain 値を設定します。

digital_gain 値は、

Mileva	0-65535 (0xffff)
--------	------------------

Mileva2/3/4	0-4095 (0xffff)
-------------	-----------------

Mileva5	0-1023 (0x3ff)
---------	----------------

の範囲です。

デフォルトの設定値は

	Mileva	Mileva2/3/4	Mileva5
greenR	0x00d4	0x010a	0x190
red	0x0117	0x017a	0x2f3
blue	0x0170	0x01ee	0x355
greenB	0x00f8	0x010e	0x190

となっています。

※現在設定されている CameraNo 側のカメラのみ値が反映されます。

※MtuOpen により値はリセットされます。

※MtuOpen にてオープンしてある必要があります。

※Mileva の設定範囲は 0-65535(0xffff)ですが、実際の範囲は 0-4095(0xffff)です

4.33 MtuXGetDigitalGain

カメラの digital_gain 値を詳細値にて読み込みます。

定義: `bool MtuGetDigitalGain (int CameraId ,WORD *greenR,WORD *red`
`,WORD *blue,WORD *greenB)`

引数:	CameraId	カメラ ID
	greenR	カメラ画素 Gr の digital_gain 値
	red	カメラ画素 R の digital_gain 値
	blue	カメラ画素 B の digital_gain 値
	greenB	カメラ画素 Gb の digital_gain 値

戻り値: true:正常 false:異常

詳細： カメラの画素の greenR=Gr red=R blue=B greenB=Gb の digital_gain 値の詳細値を読み込みます。

digital_gain 値は、

Mileva	0-65535 (0xffff)
Mileva2/3/4	0-4095 (0xffff)
Mileva5	0-1023 (0x3ff)

の範囲です。

デフォルトの設定値は

	Mileva	Mileva2/3/4	Mileva5
greenR	0x00d4	0x010a	0x190
red	0x0117	0x017a	0x2f3
blue	0x0170	0x01ee	0x355
greenB	0x00f8	0x010e	0x190

となっています。

※設定されている CameraNo 側のカメラの値です。

※MtuOpen により値はリセットされます。

※MtuOpen にてオープンしてある必要があります。

4.34 MtuXGetDigitalGainMax

カメラの digital_gain 値	詳細値	現在オープンしているカメラの最大(MAX)値を読み込みます。
---------------------	-----	--------------------------------

定義: `bool MtuGetDigitalGain (int CameraId ,WORD *greenR,WORD *red`
`,WORD *blue,WORD *greenB)`

引数:	CameraId	カメラ ID
	greenR	カメラ画素 Gr の digital_gain 値 最大(MAX) 値
	red	カメラ画素 R の digital_gain 値 最大(MAX) 値
	blue	カメラ画素 B の digital_gain 値 最大(MAX) 値
	greenB	カメラ画素 Gb の digital_gain 値 最大(MAX) 値

戻り値: true:正常 false:異常

詳細： カメラの画素の greenR=Gr red=R blue=B greenB=Gb の digital_gain 値 詳細値
における最大 (MAX) 値を読み込みます。

digital_gain 値は、

Mileva	65535 (0xffff)
Mileva2/3/4	4095 (0xffff)
Mileva5	1023 (0x3ff)

です。

※設定されている CameraNo 側のカメラの値です。

※MtuOpen にてオープンしてある必要があります。

4.35 MtuXGetDigitalGainMin

カメラの digital_gain 値	詳細値	現在オープンしているカメラの最小(MIN)値を読み込みます。
---------------------	-----	--------------------------------

定義: `bool MtuGetDigitalGain (int CameraId ,WORD *greenR,WORD *red`
`,WORD *blue,WORD *greenB)`

引数：	CameraId	カメラ ID
	greenR	カメラ画素 Gr の digital_gain 値 最小(MIN) 値
	red	カメラ画素 R の digital_gain 値 最小(MIN) 値
	blue	カメラ画素 B の digital_gain 値 最小(MIN) 値
	greenB	カメラ画素 Gb の digital_gain 値 最小(MIN) 値

戻り値: true:正常 false:異常

詳細： カメラの画素の greenR=Gr red=R blue=B greenB=Gb の digital_gain 値 詳細値
における最小(MIN)値を読み込みます。

digital_gain 値は、

Mileva	0
Mileva2/3/4	0x100
Mileva5	0

です。

※設定されている CameraNo 側のカメラの値です。

※MtuOpen にてオープンしてある必要があります。

4.36 MtuSetShutter

電子シャッタの速度の設定、変更をします。

[MtuSetExposure](#) の動作設定を細かく設定できるものです。

定義： bool MtuSetShutter (int CameraId ,WORD data)

引数： CameraId カメラ ID

 data 電子シャッタ設定値

戻り値： true:正常 false : 異常

詳細： カメラの電子シャッタの設定をします。

値は

 Mileva/2/3/4 1-65530

 Mileva5 1-10000 1=100[usec]単位

まで設定可能です。

Mileva/2/3/4

各画面サイズにより 1 カウントあたりの設定値が相違します。

各シャッタ速度の計算は下記になります。(n=シャッタ速度[sec])

USB3.0

 1280x720 60fps n / 2.222E-5 - 0.0822

 30fps n / 4.396E-5 - 0.0822

 656x408 120fps n / 1.852E-5 - 0.0987

 60fps n / 3.663E-5 - 0.0987

 328x204 240fps n / 1.852E-5 - 0.0987

 120fps n / 3.663E-5 - 0.0987

USB2.0

 1280x720 10fps n / 1.333E-4 - 0.0822

 5fps n / 2.667E-4 - 0.0822

 656x408 30fps n / 7.407E-5 - 0.0987

 15fps n / 1.481E-4 - 0.0987

 328x204 60fps n / 7.407E-5 - 0.0987

 40fps n / 1.111E-4 - 0.0987

Mileva5

USB モード/各解像度 共通

シャッタ速度 1=0.0001[sec] (0.1[msec]単位)

※現在設定されている CameraNo 側のカメラのみ値が反映されます。

※MtuOpen により値はリセットされます。

※MtuOpen にてオープンしてある必要があります。

4.37 MtuGetShutter

電子シャッタの速度の読み込みをします。

[MtuGetExposure](#) の動作設定値を細かく読み込みできるものです。

定義： bool MtuGetShutter (int CameraId ,WORD *data)

引数： CameraId カメラ ID

 *data 電子シャッタ設定値

戻り値： true:正常 false : 異常

詳細： カメラの電子シャッタの速度を取得します。

値は

 Mileva/2/3/4 1-65530

 Mileva5 1-10000 1=100[usec]単位

までです。

詳細は、[MtuSetShutter](#) を参照してください。

※現在設定されている CameraNo 側のカメラのみ値が反映されます。

※MtuOpen により値はリセットされます。

※MtuOpen にてオープンしてある必要があります。

4.38 MtuGetShutterMax

電子シャッタの速度の現在、オープン中のモデルの最大値を取得します。

[MtuSetShutter](#) / [MtuGetShutter](#) の最大値を取得します

定義: bool MtuGetShutterMax (int CameraId , WORD *data)

引数: CameraId カメラ ID

data 電子シャッタ最大値

戻り値: true:正常 false:異常

詳細: 現在オープンしているカメラの電子シャッタの最大値を取得します。
値は、

Mileva/2/3/4 65530

Mileva5 10000(1[sec])

です。

※MtuOpen にてオープンしてある必要があります。

4.39 MtuGetShutterMin

電子シャッタの速度の現在、オープン中のモデルの最小値を取得します。

[MtuSetShutter](#) / [MtuGetShutter](#) の最小値を取得します

定義: bool MtuSetShutterMin (int CameraId , WORD *data)

引数: CameraId カメラ ID

data 電子シャッタ最小値

戻り値: true:正常 false:異常

詳細: 現在オープンしているカメラの電子シャッタの最小値を取得します。
値は

Mileva/2/3/4 1

Mileva5 1(0.0001[sec])

です。

※MtuOpen にてオープンしてある必要があります。

4.40 MtuSetShutterTime

電子シャッタの速度の設定、時間単位にて変更します。

定義: bool MtuSetShutterTime (int CameraId ,double data)
引数: CameraId カメラ ID
data 電子シャッタ設定値 1[sec]～0.0001[sec] (100[usec])
戻り値: true:正常 false:異常

詳細: カメラの電子シャッタの設定を時間単位します。
値は

1[sec]～0.0001[sec] (100[usec])

まで設定可能です。

Mileva/2/3/4

DLL 内にて [MtuSetShutter](#) の計算式を使用し設定しています。

100[usec]以下は四捨五入を行うため、誤差が生じます。

100[usec]以下が設定できないサイズ・フレームレートが存在します。

その場合、DLL 内にて最小値として設定します。

下記は、シャッター速度 100[usec]を設定したときの計算値です。

USB3.0

1280x720 60fps 91[usec]

30fps 92[usec]

656x408 120fps 94[usec]

60fps 114[usec]

328x204 240fps 94[usec]

120fps 114[usec]

[MtuSetShutter](#) で100[usec]以下可能

[MtuSetShutter](#) で100[usec]以下可能

USB2.0

1280x720 10fps 144[usec]

5fps 289[usec]

100[usec]に設定できない

100/200[usec]に設定できない

656x408 30fps 81[usec]

15fps 163[usec]

100[usec]に設定できない

328x204 60fps 81[usec]

40fps 122[usec]

100[usec]に設定できない

Mileva5

1[sec]～0.0001[sec] の値

※現在設定されている CameraNo 側のカメラのみ値が反映されます。

※MtuOpen により値はリセットされます。

※MtuOpen にてオープンしてある必要があります。

4.41 MtuGetShutterTime

電子シャッタの速度、時間単位値にて取得します。

定義: `bool MtuGetShutterTime (int CameraId ,double *data)`

引数: CameraId カメラ ID

 data 電子シャッタ値 秒[sec]単位

戻り値: true:正常 false:異常

詳細: カメラの電子シャッタの設定を時間単位にて取得します。

Mileva/2/3/4

DLL 内にて [MtuSetShutter](#) の計算式を使用して計算を行ったものを返します。
計算値は、100[usec]以下の数値も含めたものです。

Mileva5

1[sec]～0.0001[sec] の値

※現在設定されている CameraNo 側のカメラのみ値が反映されます。

※MtuOpen により値はリセットされます。

※MtuOpen にてオープンしてある必要があります。

4.42 MtuGetSerialNo

シリアル No. を取得します。

定義: `bool MtuGetSerialNo(int CameraId ,DWORD *SerialNo)`

引数: CameraId カメラ ID

 *SerialNo シリアル No.

戻り値: true:正常 false:異常

詳細: シリアル No. は、4byte の 16 進数 数値 (MSB 31-bit LSB) です。

※MtuOpen にてオープンしてある必要があります。

4.43 MtuSetPixelDefectOnOff

カメラの画像に画素欠陥補正を追加します。

定義: `bool MtuSetPixelDefectOnOff(int CameraId, int onoff)`

引数: CameraId カメラ ID
onoff 0 : OFF 1 : ON (default)

戻り値: true:正常 false:異常

詳細: 1 (ON)を設定するとカメラからの画像に画素欠陥補正が追加になります。
(MtuReceiveImage で取得される画像データに画素欠陥補正された画像になります)

※MtuOpen にてオープンしてある必要があります。

※Mileva2/3/4/5 で有効です。Mileva では false:異常となります。

4.44 MtuGetPixelDefectOnOff

カメラの画素欠陥補正の状態を取得します。

定義: `bool MtuGetPixelDefectOnOff(int CameraId, int *onoff)`

引数: CameraId カメラ ID
*onoff 0 : OFF 1 : ON (default)

戻り値: true:正常 false:異常

詳細: 現在、設定されているカメラの画像に画素欠陥補正の状態を取得します。
1 (ON) の場合、画素欠陥補正が追加されています。

※MtuOpen にてオープンしてある必要があります。

※MtuOpen 後のデフォルト状態で、ON になっています。

※Mileva2/3/4/5 で有効です。Mileva では false:異常となります。

4.45 Mtu5SetAnalogGain

Mileva5 のアナログゲインを設定します。

定義： `bool Mtu5SetAnalogGain(int CameraId ,int level)`
引数： `CameraId` カメラ ID
 `level` アナログゲイン 0 ～ 1 5 (default:0)
戻り値： `true`:正常 `false` : 異常

詳細： カメラ ID におけるカメラのアナログゲインを調整します。
 0 (低) ～ 1 5 (高)

※MtuOpen にてオープンしてある必要があります。

※Mileva5 のみの動作します。その他の機種は `false` : 異常を返します

4.46 Mtu5GetAnalogGain

Mileva5 のアナログゲインを取得します。

定義： `bool Mtu5GetAnalogGain(int CameraId ,int *level)`
引数： `CameraId` カメラ ID
 `*level` アナログゲイン 0 ～ 1 5 (default:0)
戻り値： `true`:正常 `false` : 異常

詳細： 現在のカメラ ID におけるカメラのゲイン値を取得します。
 0 (低) ～ 1 5 (高)

※MtuOpen にてオープンしてある必要があります。

※Mileva5 のみの動作します。その他の機種は `false` : 異常を返します

4.47 MtuVersion

Mileva シリーズのバージョンを取得します。

定義： `bool MtuVersion(int CameraId ,char *version,int size)`

引数： `CameraId` カメラ ID
 `*version` バージョン文字列
 `size` バージョン文字列 転送最大値

戻り値： `true`:正常 `false` : 異常

詳細： Mileva シリーズのバージョンを取得します。
 バージョンは文字列で取得されます。
 サイズにてバージョンの転送容量を設定していますが、64byte 以上を設定してください。

<例>

取得されるバージョン文字列

“Version 1.00¥r¥nDate : Jul 20 2017¥r¥nTime : 15:38:17¥r¥n”

※MtuOpen にてオープンしてある必要があります。

4.48 MtuDLLVersion

本 DLL のバージョンを取得します。

定義： `bool MtuDLLVersion(char *version,int size)`

引数： `*version` バージョン文字列
 `size` バージョン文字列 転送最大値

戻り値： `true`:正常 `false` : 異常

詳細： 本 DLL のバージョンを取得します。バージョンは文字列で取得されます。
 文字列のサイズおよびバッファ容量は64byte 以上を設定してください。

<例>

取得されるバージョン文字列

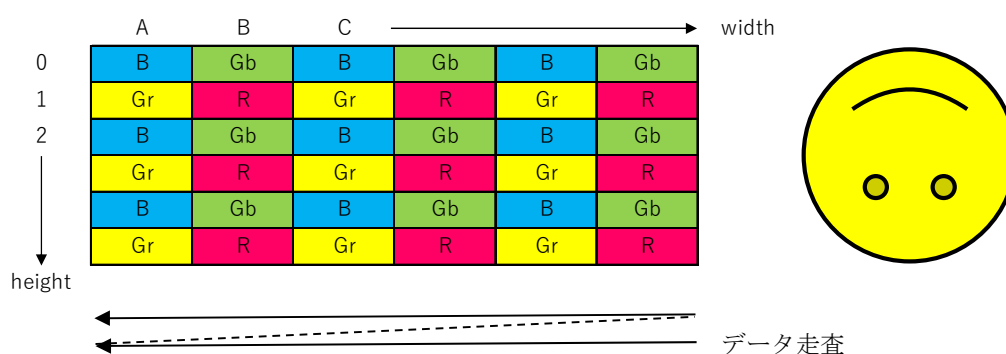
“Version 1.0.0.0¥r¥nDate : Aug 23 2017¥r¥nTime : 10:34:39¥r¥n”

5 取得画像について

MtuReceiveImage にて取得した画像は、RAW 画像になります。

下記のような配置になっているので、適時 RGB 等の画像に変換する必要があります。

5.1 RAW データ構成



データ走査は、width, height の最大値から始まります。

データ順序は、R→Gr→R→Gr … Gb→B→Gb→B … R→Gr→R→Gr となります。

5.2 画素構成

MSB										LSB						bit
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
未使用(6bit)						有効数値(10bit)										
アドレス+1								アドレス+0								

RAW データの各画素 (Gb, B, R, Gr) の有効ビット数は、各 10bit です。

実際のメモリとしては、右詰めでの各 16bit (2byte) として取り込まれます。

6 使用方法

6.1 Visual C++ 静的リンク

開発環境 Visual Studio Community 2015

6.1.1 プロジェクトの作成

[ファイル(F)]→[新規作成(N)]→[プロジェクト(P)...] を開く。

<Visual C++> → <MFC> → <MFC アプリケーション> を選択。

※インストールが必要な場合があります。

名前・場所を適時設定して <OK> を押し 次に進む。

作成内容に応じてダイアログボックス内の設定事項を行い、プロジェクトワークスペースを作成します。

6.1.2 ヘッダファイルのインクルード

Visual C++で使用するにはヘッダファイル”mtu.h” ファイルが必要となります。

ヘッダファイルのインクルードはソースコード中で記述します。

関数を使用する C++ファイルの先頭で、インクルードを指定します。

```
#include "mtu.h"
```

mtu.h の指定は C++ファイルからの相対パスで指定します。

6.1.3 ライブラリファイルのインクルード

Visual C++の場合、ヘッダファイルに加えライブラリファイルのインクルードも必要です。

[プロジェクト(P)]の[既存の項目の追加(G)...]を選択し、mtu.lib を追加してください。

その他に、下記 2 点の方法があります。

- ・プロジェクトに関連付ける方法として、プロジェクトプロパティを開き[リンカー]－[入力]－[追加の依存ファイル]として追加
- ・ソースコードに直接記述の方法として、#pragma での追加

6.2 C#

開発環境 Visual Studio Community 2015

6.2.1 プロジェクトの作成

[ファイル(F)]→[新規作成(N)]→[プロジェクト(P)...] を開く。
<Visual C#> →<Windows>→<Windows アプリケーション> を選択。
名前・場所を適時設定して <OK>を押し 次に進む。

6.2.2 ファイルの登録

Visual C#で使用するにはC#ファイル” mtu.cs” ファイルが必要となります。

[プロジェクト(P)]の[既存の項目の追加(G)...]を選択し、mtu.cs を追加してください。

6.3 OpenCV との提携方法

下記の方法により、OpenCV と提携して動作することができます。

※OpenCV および mtu.dll の開発環境への設定はできているものとします。

サンプル 画像表示(Visual C++ Win32 コンソールアプリケーション)

OpenCV バージョン 2.4.13

カメラ ID=0 , width=1280 , height=720, フレームレート=60fps 固定

```
#include "opencv2/opencv.hpp"
#include "mtu.h"
using namespace cv;

#define CAMERAID      (0)
#define WIDTH         (1280)
#define HEIGHT        (720)

int main()
{
    namedWindow("CameraView", CV_WINDOW_AUTOSIZE);
    if (MtuOpen(CAMERAID, WIDTH, HEIGHT, 60, "YUY2") != 0) {
        return 1; // エラー
    }
    int size = MtuGetBufferSize(CAMERAID); // 画像データサイズ取得
    unsigned char *buffer = new unsigned char[size]; // 画像データ保存先
    for (;;) {
        if (MtuReceiveImage(CAMERAID, buffer) == false) { // 画像データ取得
            continue;
        }
        Mat bayer16BitMat(HEIGHT, WIDTH, CV_16UC1, buffer); // 画像データをOpenCVのデータに変換
        Mat bayer8BitMat = bayer16BitMat.clone(); // 8bit ベイヤー配列データ
        bayer8BitMat.convertTo(bayer8BitMat, CV_8UC1, 0.249267); // 16bit→8bitへ変換
        Mat rgb8BitMat(HEIGHT, WIDTH, CV_8UC3); // 8bit RGB配列データ
        cvtColor(bayer8BitMat, rgb8BitMat, CV_BayerBG2BGR); // 8bitベイヤー→8bitRGBに変換
        imshow("CameraView", rgb8BitMat); // 表示
        if ((char)waitKey(10) == 27) { // ESCキー入力有無
            break;
        }
    }
    cvDestroyWindow("CameraView");
    MtuClose(CAMERAID);
    delete buffer;
    return 0;
}
```

6.4 Python サンプル

下記の方法により Python にて動作することができます

Python Version 3.14.4

OpenCV Version 4.13.0

Numpy Version 2.4.4

※OpenCV, Numpy はインストールできているものとします

※同一フォルダ内に mtu.dll, mtu.py をコピーしてください

<例> Mileva2/3/4

```
#coding: utf-8
```

```
import sys
```

```
import numpy as np
```

```
import cv2
```

```
import ctypes
```

```
from ctypes import *
```

```
import mtu
```

```
CAMERA_ID = 0
```

```
WIDTH = 1280
```

```
HEIGHT = 720
```

```
type = create_string_buffer(b"YUY2")
```

```
ret = mtu.MtuOpen(CAMERA_ID, WIDTH, HEIGHT, 60, type)
```

```
# Mtu Open
```

```
if ret != 0:
```

```
    print("MtuOpen Error")
```

```
    sys.exit()
```

```
imgsize = mtu.MtuGetBufferSize(CAMERA_ID)
```

```
# 画像データサイズ取得
```

```
raw = create_string_buffer(imgsize)
```

```
# 画像データ保存エリア
```

```
img2 = np.empty((HEIGHT, WIDTH), dtype=np.uint8)
```

```
# 8bit ベイヤー変換先エリア
```

```
while True:
```

```
    ret = mtu.MtuReceiveImage(CAMERA_ID, raw)
```

```
# 画像取得 10bit ベイヤー
```

```
    if ret == False:
```

```
        continue
```

```
    img1 = np.frombuffer(raw, dtype=np.uint16).reshape(HEIGHT, WIDTH) # c_char->numpy uint16 変換
```

```
    for h in range(0, HEIGHT):
```

```
        for w in range(0, WIDTH):
```

```
            img2[h][w] = img1[h][w]>>2
```

```
# 10bit ベイヤー->8bit ベイヤー変換
```

```
    img3 = cv2.cvtColor(img2, cv2.COLOR_BayerBG2BGR);
```

```
# 8bit ベイヤー->8bitRGB に変換
```

```
    cv2.imshow('CameraView', img3)
```

```
# 画像表示
```

```
    if cv2.waitKey(10) == 27:
```

```
# esc で終了
```

```
        break
```

```
cv2.destroyAllWindows()
```

```
# 画像閉じる
```

```
mtu.MtuClose(CAMERA_ID);
```

```
# Mtu Close
```

```
sys.exit()
```

7 注意事項

7.1 色空間タイプ（メディアサブタイプ）

本機器ではデバイスドライバとして、UVC(USB Video Class)を使用しています。

本来規定のメディアサブタイプを使用しますが、本機器は RAW 画像出力のため

MEDIASUBTYPE_YUY2（YUY2 サブタイプ）

を使用し、対応しています。

その為、UVC 対応の通常のアプリケーションソフトでは緑色画面表示となります。

7.2 他の UVC カメラの動作

他の UVC 対応カメラの一部の YUV2 製品は動作しますが、RAW 画像出力仕様のため基本的にはサポートしていません。

8 Mileva シリーズの相違点

Mileva の機種ごとに下記の相違点があります。

	Mileva	Mileva2	Mileva3	Mileva4	Mileva5
形状	うちわ型	円筒型	角型	長方形	角型
デバイス名	MTU	MTU2	MTU2	MTU2	MTU5
GPIO	あり	なし	なし	なし	なし
照明(LED)	内部	内部	外部	内部	外部
画素欠陥補正	なし	あり	あり	あり	あり
最大画面表示	1280x720	1280x720	1280x720	1280x720	1920x1080

※画素欠陥補正機能は mtu.dll Version 1.1.0.0 以降に有効

※Mileva5 機能は mtu.dll Version 1.2.0.0 以降に有効

株式会社クリオ

〒491-0201 愛知県一宮市奥町字神田 19-1

TEL:0586-64-3177

FAX:0586-64-3199

URL: <https://www.cli-o.com>