Mileva シリーズ
Software Developer Kit
MX-02

株式会社イチワ

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

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

MtuSetExpsoure 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

目次

本書に関する注意事項

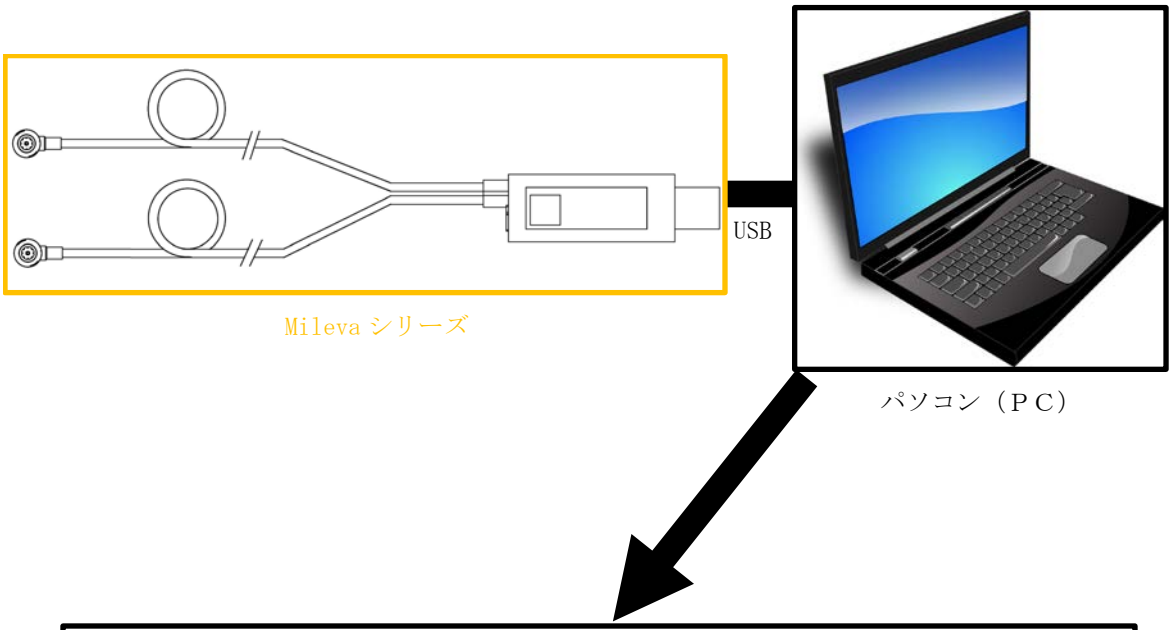
1．ご使用の前に必ず本書をよくお読みになり、注意事項を確認のうえ製品を正しくご使用ください。
2．本書は必要なときに参照できるよう、大切に保管してください。
3．本製品を本来の目的以外の用途でお使いになった場合には、安全性を保証致しかねますので、 ご了承ください。
4．本書の内容に関して、将来予告なしに変更することがあります。
5．本書に掲載している図は、説明のため、一部、省略や抽象化を行っています。
6．本書の内容に関して、万一ご不審な点や誤り、記載漏れなどお気づきのことがございましたら、本書末尾記載の弊社連絡先または販売店までご連絡ください。
7．本書の内容の一部または全部を、弊社に無断で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

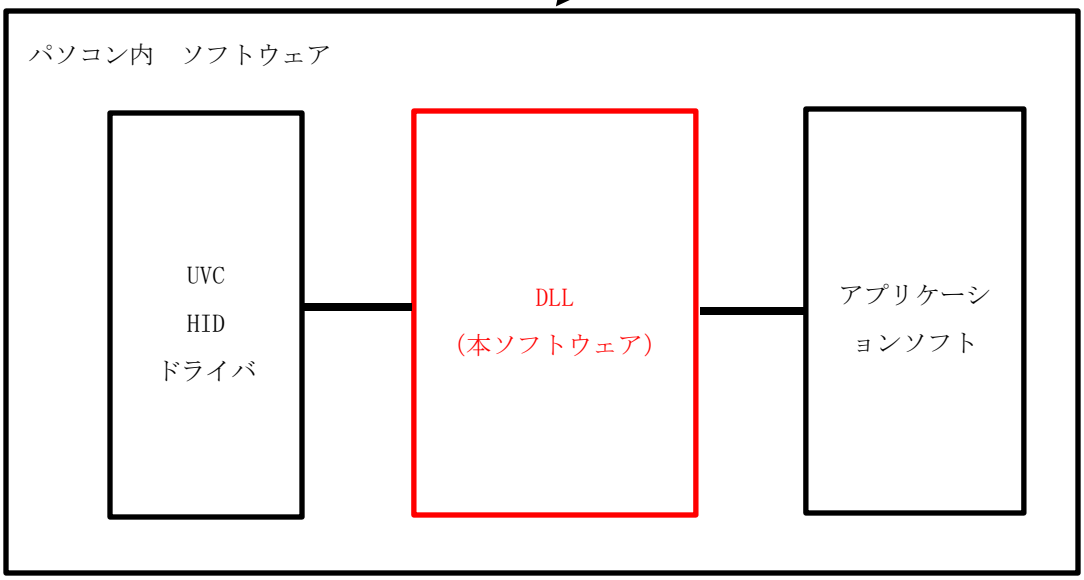
弊社では、本製品を運用した結果での損失、逸失利益等の請求につきましては、項目６に関わらずいかなる責任も負いかねますので、予めご了承ください。

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

Windows10　32bit/64bit

Windows8　32bit/64bit

Windows7　32bit/64bit

2.2 対応開発言語

C++	Visual Studio Community 2017 にて動作確認済み
C#.NET	Visual Studio Community 2017 にて動作確認済み
VC.NET	
VB.NET	
Delphi	

2.3 ライブラリ構成

Release note.txt	改版履歴
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
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

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

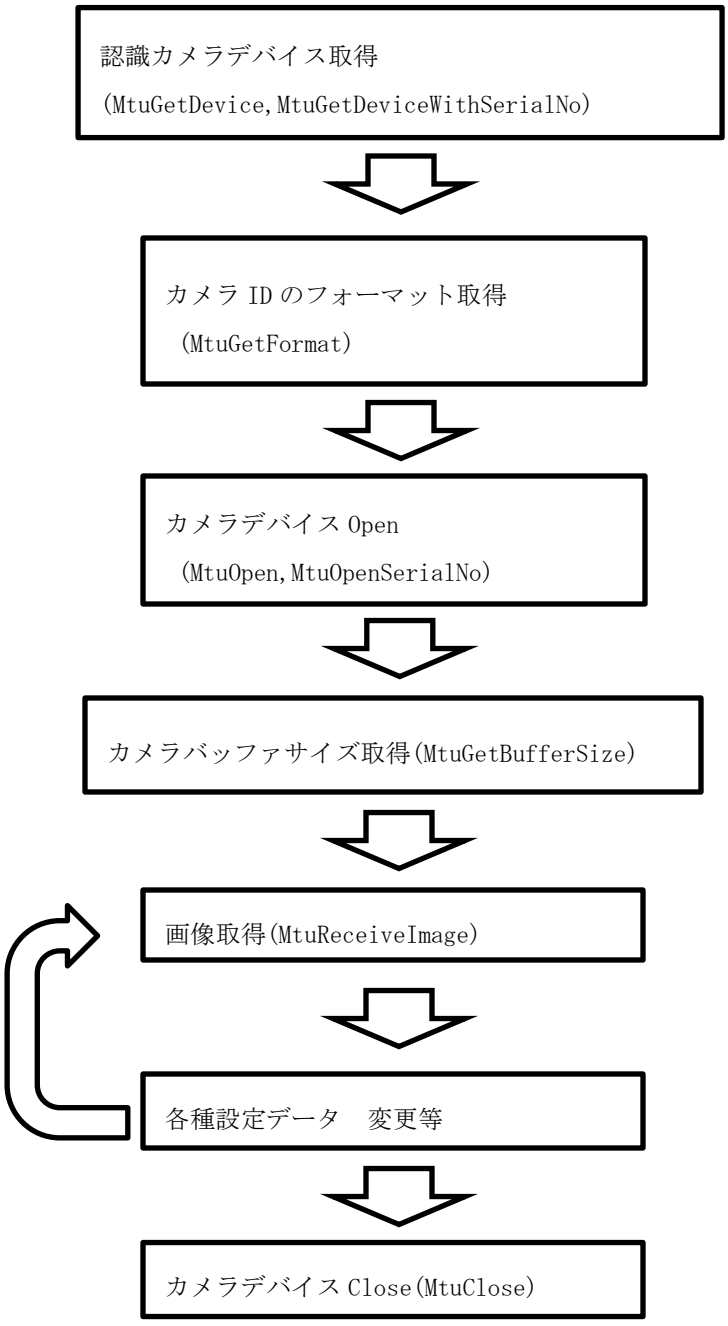
本 DLL は下記の機種に対応しています。

Mileva Sensor	MS-011	１カメラモデル
	MS-012	２カメラモデル
Mileva2 Sensor	MS-111	１カメラモデル(カメラ　側視モデル)
	MS-211	１カメラモデル(カメラ　直視モデル)

Mileva シリーズは、上記の全モデルを示します。

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
MtuOpen	オープン
MtuOpenSerialNo	オープン(シリアル No.)
MtuClose	クローズ
MtuGetBufferSize	バッファサイズ取得
MtuGetDevice	カメラ認識デバイス取得
MtuGetDeviceWithSerialNo	カメラ認識デバイス取得（シリアル No.付)
MtuGetFormat	カメラフォーマット取得
MtuReceiveImage	画像データ受信
MtuSetLed	照明設定
MtuGetLed	照明取得
MtuSetGain	ゲイン設定
MtuGetGain	ゲイン取得
MtuSetExposure	露出設定
MtuGetExposure	露出取得
MtuSetCameraNo	カメラ No.設定
MtuGetCameraNo	カメラ No.取得
MtuSetColorBar	カラーバー設定
MtuGetColorBar	カラーバー取得
MtuCameraRecognition	カメラ認識取得
MtuUSBMode	USB モード取得
MtuGPIODirection	GPIO 入出力設定
MtuGPIOOutput	GPIO 出力設定
MtuGPIOInput	GPIO 入力
MtuTemperature	温度取得
MtuSetDigitalGain	digital_gain の設定
MtuGetDigitalGain	digital_gain の読込

関数名	説明
MtuSetShutter	電子シャッタ設定
MtuGetShutter	電子シャッタ読込
MtuGetSerialNo	シリアル No 読込
MtuSetPixelDefectOnOff	画素欠陥　ON/OFF 設定
MtuGetPixelDefectOnOff	画素欠陥　ON/OFF 読込
MtuSetShadingCorrectionOnOff	シェーディング補正　ON/OFF 設定
MtuGetShadingCorrectionOnOff	シェーディング補正　ON/OFF 読込
MtuVersion	バージョン情報読込
MtuDLLVersion	DLL バージョン情報読込

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

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

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#”もしくは”MTU2#”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

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

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 1.1.0.0以降､機種コード “MTU#”もしくは”MTU2#”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

1280x720x2	⇒	1,843,200 byte
656x408x2	⇒	535,296 byte
328x204x2	⇒	133,824 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 Sensor の場合、”MTU#xxxxxxxx”(x=シリアル No 数値 8 桁)となります。

Mileva2 Sensor の場合、”MTU2#xxxxxxxx”(x=シリアル No 数値 8 桁)となります。

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 では、画素欠陥補正、シェーディング補正が追加(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　０～６４(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）～６４（明）

※この照明(LED)は両カメラで共通です。（両方、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　０～６４(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）～６４（明）

※この照明(LED)は両カメラで共通です。（両方、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　０～２４０(default 120)
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）～２４０（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。

4.13 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　０～２４０(default 120)
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）～２４０（高）

※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。

4.14 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

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)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。

　1280x720　　5fps　　露出時間 -13　　-12 の値を設定

4.15 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　－１３～０

戻り値：　　true:正常 false：異常

詳細：　　現在のカメラ ID におけるカメラの露出の値を取得します。

　　―１３（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。

4.16 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。

4.17 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。

4.18 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。

4.19 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。

4.20 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。

4.21 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。

4.22 MtuGPIODirection

外部接続ＧＰＩＯ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ＧＰＩＯポートの入出力を設定します。

　　GPIO0　　0bit　　０：入力 １：出力

　　GPIO1　　１bit　　０：入力 １：出力

　　GPIO2　　２bit　　０：入力 １：出力

　　GPIO3　　３bit　　０：入力 １：出力

　　GPIO4　　４bit　　０：入力 １：出力

　　GPIO5　　５bit　　０：入力 １：出力

　　GPIO6　　６bit　　０：入力 １：出力

　　GPIO7　　７bit　　０：入力 １：出力

　　＜例＞

　　GPIO1,2,4,6,7 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 Sensor のみ対応。その他の機種では false:異常になります。

4.23 MtuGPIOOutput

外部接続ＧＰＩＯ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ＧＰＩＯポートの出力を設定します。

　　GPIO0　０bit　０：Lo　１：Hi

　　GPIO1　１bit　０：Lo　１：Hi

　　GPIO2　２bit　０：Lo　１：Hi

　　GPIO3　３bit　０：Lo　１：Hi

　　GPIO4　４bit　０：Lo　１：Hi

　　GPIO5　５bit　０：Lo　１：Hi

　　GPIO6　６bit　０：Lo　１：Hi

　　GPIO7　７bit　０：Lo　１：Hi

　　＜例＞

　　GPIO1,2,4 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 Sensor のみ対応。その他の機種では false:異常になります。

4.24 MtuGPIOInput

外部接続ＧＰＩＯ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ＧＰＩＯポートの入力値を取得します。

　　GPIO0　　0bit　　０：Lo　１：Hi

　　GPIO1　　１bit　　０：Lo　１：Hi

　　GPIO2　　２bit　　０：Lo　１：Hi

　　GPIO3　　３bit　　０：Lo　１：Hi

　　GPIO4　　４bit　　０：Lo　１：Hi

　　GPIO5　　５bit　　０：Lo　１：Hi

　　GPIO6　　６bit　　０：Lo　１：Hi

　　GPIO7　　７bit　　０：Lo　１：Hi

　　＜例＞

　　GPIO1,2,4 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 Sensor のみ対応。その他の機種では false:異常になります。

4.25 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、－５５℃　～　＋１２８℃　迄の値です。

※MtuOpen にてオープンしてある必要があります。

4.26 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(0-65535)
	red	カメラ画素 R の digital_gain 値(0-65535)
	blue	カメラ画素 B の digital_gain 値(0-65535)
	greenB	カメラ画素 Gb の digital_gain 値(0-65535)

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、0 ～ 65535 迄の値です。

デフォルトの設定値は

	Mileva	Mileva2
greenR	0x00d4	0x012a
red	0x0117	0x019a
blue	0x0170	0x019a
greenB	0x00f8	0x00ba

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。

4.27 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(0-65535)
red		カメラ画素 R の digital_gain 値(0-65535)
blue		カメラ画素 B の digital_gain 値(0-65535)
greenB		カメラ画素 Gb の digital_gain 値(0-65535)

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、0 ～ 65535 迄の値です。

デフォルトの設定値は

	Mileva	Mileva2
greenR	0x00d4	0x012a
red	0x0117	0x019a
blue	0x0170	0x019a
greenB	0x00f8	0x00ba

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。

4.28 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(1-65530)

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 1-65530 まで設定可能です。

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

※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。

4.29 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(1-65530)

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 1-65530 まで取得可能です。

　　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。

4.30 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 31-0bit LSB)です。

※MtuOpen にてオープンしてある必要があります。

4.31 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

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 で有効です。(Mileva では false を返します)

4.32 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

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 で有効です。(Mileva では false を返します)

4.33 MtuSetShadingCorrectionOnOff

カメラの画像にシェーディング補正を追加します。

定義：　　bool **MtuSetShadingCorrectionOnOff**(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 で有効です。(Mileva では false を返します)

4.34 MtuSetShadingCorrectionOnOff

カメラのシェーディング補正の状態を取得します。

定義：　　bool **MtuGetShadingCorrectionOnOff**(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、設定されているカメラの画像にシェーディング補正の状態を取得します。

　　1(ON)の場合、シェーディング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 で有効です。(Mileva では false を返します)

4.35 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。

4.36 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

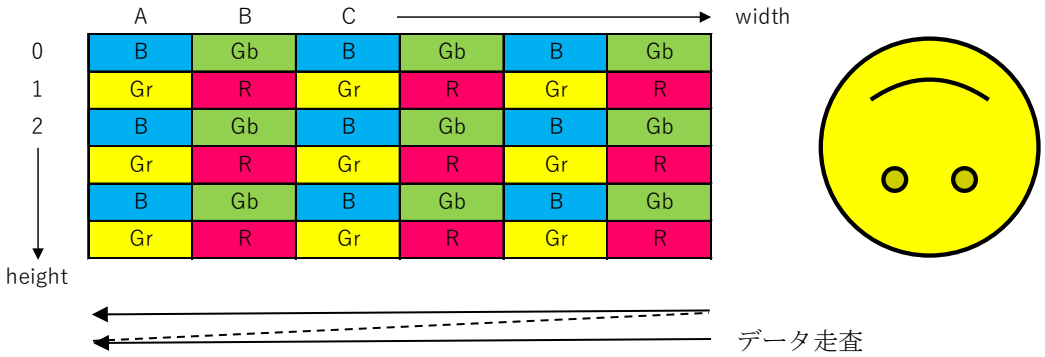
“Version 1.0.0.0¥r¥nDate ： Aug 23 2017¥r¥nTime ： 10:34:39¥r¥n”

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。

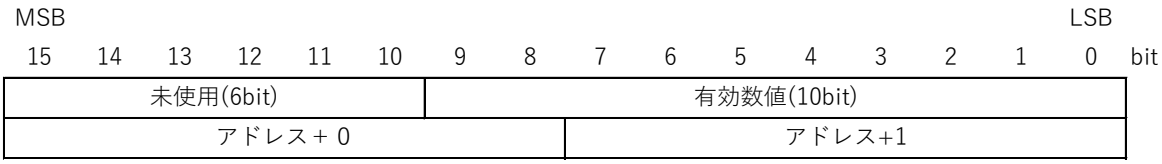
5.1　RAW データ構成



データ走査は、width,heiht 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



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)]→[ﾌﾟﾛｼﾞｪｸﾄ(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)]→[ﾌﾟﾛｼﾞｪｸﾄ(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); // 画像ﾃﾞｰﾀをOpenCVのﾃﾞｰﾀに変換
        Mat bayer8BitMat = bayer16BitMat.clone();         // 8bit ベイヤー配列データ
        bayer8BitMat.convertTo(bayer8BitMat, CV_8UC1, 0.249267); // 16bit→8bitﾍﾞｲﾔ変換
        Mat rgb8BitMat(HEIGHT, WIDTH, CV_8UC3);           // 8bit RGB配列データ
        cvtColor(bayer8BitMat, rgb8BitMat, CV_BayerBG2BGR); // 8bitﾍﾞｲﾔｰ→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;
}
```

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 Sensor (MS-012,MS-011)	Mileva2 Sensor (MS-111,MS-211)
デバイス名	MTU	MTU2
GPIO	あり	なし
画素欠陥補正	なし	あり
シェーディング補正	なし	あり

※画素欠陥補正及びシェーディング補正機能は mtu.dll Version 1.1.0.0 以降に有効

株式会社イチワ

〒491-0201　　愛知県一宮市奥町字神田 19-1

TEL:0586-64-2191

FAX:0586-64-2900

URL: http://www.ichiwa.com