

[EVT-AN04-BIRGER]

EMERGENT VISION TECHNOLOGIES INC 1.0.1

Dec 1, 2014

SUITE #239 - 552A CLARKE ROAD, COQUITLAM, V3J 0A3, B.C. CANADA

WWW.EMERGENTVISIONTEC.COM



連絡先（日本国代理店）

株式会社アルゴ

〒532-0003 大阪府大阪市淀川区宮原 1 丁目 2-4 新大阪第 5 ドイビル 7F

販売および技術相談窓口（日本語）

argo@argocorp.com

06-6151-9283

<https://www.argocorp.com/>

Legal

Life Support Applications

These products are not designed for use in life support systems.

Trademarks

All trademarks appearing in this document are protected by law.

Warranty

The information provided is supplied without any guarantees or warranty.

Copyright

All texts, pictures, files, and graphics are protected by copyright and other laws protecting intellectual property. It is not permitted to copy or modify them for and use.



概要

このアプリケーションノートでは、EVT のカメラが電子アイリス制御と電子フォーカス制御に BIRGER マウントアダプタを使用するメカニズムについて説明します。下の写真は、EVT_BG アダプターと標準の Canon EF レンズに接続された HR-12000 です。



このようなソリューションの利点は、メインのカメラインターフェイス（この場合は 10GigE）を介して、ソフトウェアでレンズを制御できる事です。

10GigE SFP +インターフェイスは、ダイレクトアタッチケーブルは 10 m 以下、又はファイバーオプションを使用した場合のケーブル長は 1 km を超えます。このレンズのソフトウェア制御は、ケーブル長に関係なく可能です。

左下の画像は、PWR_BREAK_BIRGER です。PWR_BREAK_BIRGER は、カメラへの電源供給用コネクタ（12 ピン）、EVT_BG モジュールへの電源&フォーカスアイリスライン（4 ピン）、外部トリガーなどの一般的な GPIO 用信号ピン（バラ線）、の 3 つがあります。右の写真は、4 ピン Hirose コネクタを介して電氣的に接続された EVT_BG モジュールを示しています。あとは、EVT_BG マウントを該当するカメラの M42、M52、または C マウント前面に接続し、上記のように 12 ピン Hirose コネクタをカメラの背面に差し込むだけです。これでハードウェアの接続は完了です。



導入

EVT_BG アダプターは、以下の型番で株式会社アルゴより提供しています。

型番 : EVT_BG_M52/CanonEF

商品名 : EVT アイリスフォーカス制御 M52/EF マウント

詳細 : EVT カメラ用 M52/EF 変換アダプタ 12VTTL 9600 ボーレート

型番 : EVT_BG_M42/CanonEF

商品名 : EVT アイリスフォーカス制御 M42/EF マウント

詳細 : EVT カメラ用 M42/EF 変換アダプタ 12VTTL 9600 ボーレート

型番 : EVT_BG_C/CanonEF

商品名 : EVT アイリスフォーカス制御 C/EF マウント

詳細 : EVT カメラ用 C/EF 変換アダプタ 12VTTL 9600 ボーレート

型番 : PWR_BREAK_BIRGER

商品名 : パワーサプライ&バーシヤマウント&GPIO

詳細 : EVT_BG4ピン/カメラ 12ピン/バラ線 GPIO

例えば、CマウントのEVT社のカメラを使う場合は、EVT_BG_C/CanonEFとPWR_BREAK_BIRGERをご注文下さい。

- 光学制御アダプターは 9600 ボーレートでコントロール
- 4ピンケーブルは 150mm
- 4ピンのピンアサイン : 1.Ground, 2.Transmit, 3.Receive, 4.Power
- アダプタのコネクタは Hirose HR10A-7P-4P (73)
- 相手方のコネクタは Hirose HR10A-7J-4S (73)
- バラ線の GPO3 と GPI5 は EVT_BG アダプターに割り当てられています。他の IO はトリガー等の利用可
- 電氣的には EVT_BG_M52/ CanonEF と EVT_BG_C/ CanonEF と EVT_BG_M42/ CanonEF はそれぞれ互換性はありません。
メカニカル的には EVT_BG_C/ CanonEF と EVT_BG_M42/ CanonEF は互換性がありますが、EVT_BG_M52/ CanonEF は互換性はありません。

ソフトウェア側では、Emergent の eSDK ソフトウェアを介して EVT_BG へのアクセスします。次のソースコードは、モジュールを使うプログラムの例です。Norpix は EVT のカメラを StreamPIX でサポートしました。従って、レンズコントロールは主要なソフトウェアで制御できます。

使用可能なコマンドや各種ソフトウェアのマニュアルは、以下よりダウンロード可能です。

<https://birger.com/download/>

eSDK には EVT_GPIO の例があります。Extras / Birger_Release の FTP で .cpp ファイルを EVT_GPIO.cpp ファイルに置き換えて再コンパイルすると、コマンドを送信して EVT_BG とレンズから応答を受信できるコンソールプログラムが表示されます。EVT_GPIO を入力して実行します。

コンソール画面で、「*」を入力して、受信 Uart バッファータを読み取ります。すべてのコマンドの後に実行できます。ビット深度は 256 です。「!」でプログラムを終了します。

コンソール画面からは、通常のシリアルコンソールから行うように、Birger コマンドを入力できます。
(<>は含めない) :

```
<enter> //First enter clears transmit buffer
<enter>
<*> //Read back buffer which has various outputs from Birger unit.
<in> //Initialize aperture
<*>
<mc> //Iris close all the way
<*>
<mo> //Iris open all the way
<*>
<mi> //Move focus to infinity stop
<*>
<mz> //Move focus to zero stop
<*>
<!> //Quit
```

Here is the partial source code for the example:

```
//Use GPO3 as Uart transmit
EVT_CameraSetEnumParam(&camera, "GPO_3_Mode", "Test_Gen_Uart_Txd");

//Enables receive Uart which will be using GPI5.
EVT_CameraSetBoolParam(&camera, "UartEnable", TRUE);

//Set Uart baud rate for tx and rx
EVT_CameraSetEnumParam(&camera, "UartBaud", "B_9600");

//Set Uart number of data bits.
EVT_CameraSetUInt32Param(&camera, "UartDataBits", 8);

//Set Uart number of stop bits.
EVT_CameraSetUInt32Param(&camera, "UartStopBits", 1);

user_char = _getch();
uart_data = user_char & 0xFF;
while(user_char!='!') //'h' to start as set above.
{
    if(user_char == '*') //We want to read all that is in the uart recv buffer.
    {
        unsigned int uart_rx_cnt, uart_rx_data;
        printf("\n*****Reading Rx Fifo Data*****\n");
        EVT_CameraGetUInt32Param(&camera, "UartRxFifoCnt", &uart_rx_cnt);
        for(int i=0;i<uart_rx_cnt;i++)
        {
            EVT_CameraGetUInt32Param(&camera, "UartRxData", &uart_rx_data);
            uart_rx_data = uart_rx_data & 0xFF;
            if(uart_rx_data == 0x0D)
                printf("\n");
            else
                printf("%c", uart_rx_data);
        }
        printf("\n*****\n");
    }
#ifdef _MSC_VER
    else if(user_char == 0x0D) //CR
#else
    else if(user_char == 0x0A) //LF
#endif
    {
        EVT_CameraSetUInt32Param(&camera, "UartTxData", 0x0D); //CR
        EVT_CameraSetUInt32Param(&camera, "UartTxData", 0x0A); //LF
        printf("\n");
    }
    else
```

```
{  
    EVT_CameraSetUInt32Param(&camera, "UartTxData", uart_data);  
    printf("%c", user_char);  
}  
  
user_char = _getch();  
  
uart_data = user_char & 0xFF;  
}
```