

## **[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](http://WWW.EMERGENTVISIONTEC.COM)

Authorized Distributor:



*Your single source for Imaging Solutions*

Ph. 510.657.4000

[info@uniforcesales.com](mailto:info@uniforcesales.com)

[www.uniforcesales.com](http://www.uniforcesales.com)



## Contact

### Emergent Vision Technologies Canada (Headquarters)

Suite #239 - 552A Clarke Road,

Coquitlam, B.C.

V3J 0A3

CANADA

[info@emergentvisiontec.com](mailto:info@emergentvisiontec.com)

[www.emergentvisiontec.com](http://www.emergentvisiontec.com)

### Technical Support

[info@emergentvisiontec.com](mailto:info@emergentvisiontec.com)

### Authorized Distributor:



*Your single source for Imaging Solutions*

Ph. 510.657.4000

[info@uniforcesales.com](mailto:info@uniforcesales.com)

[www.uniforcesales.com](http://www.uniforcesales.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.



## Overview

This application note describes the mechanism by which Emergent cameras can use Birger Engineering Canon EF adapters for electronic iris and focus control. Below is pictured the HS-12000 connected to the Birger Canon EF adapter and a standard Canon EF lens.



The benefits of such a solution are to be able to bring the lens under software control with all communications happening over the main camera interface - in this case, the enormously fast 10GigE interface. As we have noted in the past, the 10GigE SFP+ interface has ultimate flexibility for cable lengths with direct attach ( $\leq 10\text{m}$ ) and fiber options to over a kilometer. This application note applies regardless of cable length.

On the left below is Emergent accessory PWR\_BREAK\_BIRGER which supplies power to the camera, breaks out 4 pins for Birger module use, and the balance of the signals is broken out for general GPIO use such as external triggering. The picture on the right shows now the Birger module electrically connected via the 4 pin Hirose connector. All that is left is to screw the Birger mount into the M42, M52, or C-mount front of the applicable camera and then plug the 12 pin Hirose connector into the back of the camera as shown above. This completes the hardware installation.



## Implementation

The Birger adapters are ordered directly from Birger Engineering with the part numbers below:

Camera: HS-20000

Part: RD-EF1-EV2-EV1

Description: Emergent M52 Canon EF Lens Interface (M52x1.5, 12V TTL-UART Cable, 9600 Baud)

Camera: HS-2000, HS-4000

Part: RD-EF1-CM0-EV1

Description: Emergent C-Mount Canon EF Lens Interface (C-Mount, 12V TTL-UART Cable, 9600 Baud)

Camera: HS-12000

Part: RD-EF1-DG2-EV1

Description: Emergent M42 Canon EF Lens Interface (M42x1, 12V TTL-UART Cable, 9600 Baud)

### The accessories provided by Emergent:

Part: PWR\_BREAK\_BIRGER (Power supply with breakout for Birger, flying leads for rest of GPIO)  
(for those still needing additional IO for say external triggering)

Part: PWR\_BIRGER (Power supply with breakout for Birger)  
(for those not needing additional IO)

So, for the HS-12000, for example, you might order PWR\_BREAK\_BIRGER and RD-EF1-DG2-EV1 and this will provide a complete solution as in the picture above.

### Additional Notes:

- Optimal control of the adapter is through a baud rate of 9600.
- The 4 pin cable length is 150mm.
- The pinout of the 4 pin connector: 1. Ground, 2. Transmit, 3. Receive, 4. Power
- The adapter connector is Hirose HR10A-7P-4P(73)
- The mating 4 pin connector on Emergent accessories is Hirose HR10A-7J-4S(73)
- Camera IO GPO3 and GPI5 are reserved for the Birger adapter when using the above Emergent accessories. Other IO still available for triggering, etc.
- Please contact Birger for retrofit of existing Birger adapters (no others are directly compatible). Electrically, no others are compatible. Mechanically, some compatibility can be seen with RD-EF1-CM0-EV1 and RD-EF1-DG2-EV1 but not with RD-EF1-EV2-EV1.



On the software side, we provide access to the Birger adapter through Emergent's eSDK software. The source code on the next page shows how a program can be written to speak with the module. Norpix has added full support in their software so lens control is through main software interface.

Full functional example source code is available on our FTP site in the directory Extras/Birger\_Release. Please request access to the FTP site through support at [info@emergentvisiontec.com](mailto:info@emergentvisiontec.com).

The Birger adapter manual which describes the commands available for the adapter can be found here: [http://www.birger.com/pdf/ef232\\_user\\_manual.pdf](http://www.birger.com/pdf/ef232_user_manual.pdf)

In our eSDK there is an example EVT\_GPIO. If you replace the .cpp file with the EVT\_GPIO.cpp file on our FTP in Extras/Birger\_Release and recompile you get a console program that allows you to send commands and receive replies from the Birger adapter and lens.

Run by typing EVT\_GPIO.

Type h to enter console.

Type \* to read receive Uart buffer data. Can do after every command. Buffers are 256 deep.

Type ! to exit program.

From the console you can enter Birger commands as you would from a regular serial console. As a simple test after you type h(don't include <>):

```
<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 rcv 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;
}
```



## Document History

Version	Date	Description
1.01	1 Dec 2014	Initial Version

