

TECHNICAL REFERENCE

FLIR **BLACKFLY**[®]

USB3 Vision



USB[™]
VISION

Version 6.2
Revised 8/30/2018



FCC Compliance

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesirable operation.

Korean EMC Certification

The KCC symbol indicates that this product complies with Korea's Electrical Communication Basic Law regarding EMC testing for electromagnetic interference (EMI) and susceptibility (EMS).

Hardware Warranty

The warranty for the Blackfly USB 3.1 camera is 3 years. For detailed information on how to repair or replace your camera, please see the [terms and conditions on our website](#).

WEEE

The symbol indicates that this product may not be treated as household waste. Please ensure this product is properly disposed as inappropriate waste handling of this product may cause potential hazards to the environment and human health. For more detailed information about recycling of this product, please contact us.



Trademarks

Names and marks appearing on the products herein are either registered trademarks or trademarks of FLIR Systems, Inc. and/or its subsidiaries.

Licensing

To view the licenses of open source packages used in this product please see [What open source packages does firmware use?](#)



Table of Contents

About This Manual	i
1 Blackfly USB 3.1 Specifications	1
1.1 BFLY-U3-03S2 Specifications	1
1.2 BFLY-U3-05S2 Specifications	2
1.3 BFLY-U3-13S2 Specifications	3
1.4 BFLY-U3-20S4 Specifications	4
1.5 BFLY-U3-23S6 Specifications	5
1.6 BFLY-U3-50H5 Specifications	6
1.7 Analog-to-Digital Converter (ADC)	7
2 Blackfly USB 3.1 Installation	8
2.1 Before You Install	8
2.1.1 Will your system configuration support the camera?	8
2.1.2 Do you have all the parts you need?	8
2.1.3 Do you have a downloads account?	8
2.2 Installing Your Interface Card and Software	9
2.3 Installing Your Camera	10
2.4 Configuring Camera Setup	11
2.4.1 Configuring Camera Drivers	11
2.4.2 Maximum Number of Cameras on a Single Bus	11
3 Tools to Control the Blackfly USB 3.1	12
3.1 Using Spinnaker®	12
3.1.1 SpinView Program	12
3.1.2 Custom Applications Built with the Spinnaker API	12
3.2 Using FlyCapture®	12
3.2.1 FlyCap Program	12
3.2.2 Custom Applications Built with the FlyCapture API	13
3.3 Using GenICam Applications	13
3.4 Using Control and Status Registers	13
4 Blackfly USB 3.1 Physical Interface	15
4.1 Blackfly USB 3.1 Physical Description	15
4.2 Blackfly USB 3.1 Dimensions	16

4.3 Mounting with the Case or Mounting Bracket	17
4.4 Lens Mounting	17
4.4.1 Back Flange Distance	18
4.5 Handling Precautions and Camera Care	19
4.5.1 Case Temperature and Heat Dissipation	19
4.6 Dust Protection	20
4.7 Infrared Cut-Off Filters	21
4.8 Camera Interface and Connectors	22
4.8.1 USB 3.1 Connector	22
4.8.2 Interface Cables	22
4.8.3 Interface Card	23
4.8.4 General Purpose Input/Output (GPIO)	23
5 General Blackfly USB 3.1 Operation	24
5.1 Powering the Camera	24
5.2 User Sets (Memory Channels)	25
5.2.1 GenICam User Set Control	25
5.3 On-Camera Frame Buffer	26
5.4 Non-Volatile Flash Memory	26
5.5 Camera Firmware	27
5.5.1 Determining Firmware Version	27
5.5.2 Upgrading Camera Firmware	27
6 Input/Output Control	29
6.1 General Purpose Input/Output (GPIO)	29
6.2 GenICam Digital Input/Output Control	30
6.3 Strobe Signal Delay	31
6.4 Programmable Strobe Output	32
6.5 Debouncer	33
6.6 GPIO Electrical Characteristics	34
6.6.1 Output Timing Characteristics	35
6.6.2 Input Timing Characteristics	37
7 Image Acquisition	39
7.1 Asynchronous Triggering	39
7.1.1 GenICam Acquisition Control	39
7.1.2 Standard External Trigger (Mode 0)	41

7.1.3 Bulb Shutter Trigger (Mode 1)	42
7.1.4 Low Smear Trigger (Mode 13)	43
7.1.5 Overlapped Exposure Readout Trigger (Mode 14)	44
7.1.6 Multi-Shot Trigger (Mode 15)	45
7.2 External Trigger Timing	46
7.3 Camera Behavior Between Triggers	46
7.4 Changing Video Modes While Triggering	47
7.5 Asynchronous Software Triggering	48
8 Blackfly USB 3.1 Attributes	49
8.1 Pixel Formats	49
8.1.1 Raw	49
8.1.2 Mono	49
8.1.3 RGB	49
8.1.4 YUV	49
8.2 Video Modes Overview	51
8.2.1 Blackfly USB 3.1 Video Mode Descriptions	52
8.3 GenICam Image Format Control	54
8.4 Frame Rates	56
8.4.1 BFLY-U3-03S2 Frame Rates by Video Mode	56
8.4.1.1 BFLY-U3-03S2M	56
8.4.1.2 BFLY-U3-03S2C	56
8.4.2 BFLY-U3-05S2 Frame Rates by Video Mode	57
8.4.2.1 BFLY-U3-05S2M	57
8.4.2.2 BFLY-U3-05S2C	57
8.4.3 BFLY-U3-13S2 Frame Rates by Video Mode	58
8.4.3.1 BFLY-U3-13S2M	58
8.4.3.2 BFLY-U3-13S2C	58
8.4.4 BFLY-U3-20S4 Frame Rates by Video Mode	59
8.4.4.1 BFLY-U3-20S4M	59
8.4.4.2 BFLY-U3-20S4C	59
8.4.5 BFLY-U3-23S6 Frame Rates by Video Mode	60
8.4.5.1 BFLY-U3-23S6M	60
8.4.5.2 BFLY-U3-23S6C	60
8.4.6 BFLY-U3-50H5 Frame Rates by Video Mode	61

8.4.6.1 BFLY-U3-50H5M	61
8.4.6.2 BFLY-U3-50H5C	61
8.5 Readout Method (Shutter Type)	62
8.5.1 Global Shutter	62
8.6 Overview of Imaging Parameters	63
8.7 GenICam Analog Control	64
8.8 Brightness	65
8.9 Exposure Time	65
8.10 Gain	66
8.11 Auto Exposure	67
8.12 Sharpness	68
8.13 Gamma and Lookup Table	68
8.14 High Dynamic Range (HDR) Imaging	70
8.15 Image Flip/Mirror	70
8.16 Embedded Image Information	70
8.17 White Balance	72
8.18 Bayer Color Processing	73
8.19 Hue	75
8.20 Saturation	75
9 Troubleshooting	76
9.1 Downloads and Support	76
9.1.1 Finding Information	76
9.1.2 Contacting Technical Support	76
9.2 Camera Diagnostics	77
9.3 Status Indicator LED	77
9.4 Test Pattern	78
9.5 Channel Balancing	79
9.6 Link Recovery	80
9.7 Blemish Pixel Artifacts	81
9.7.1 Pixel Defect Correction	81
9.8 Vertical Smear Artifact	82
9.8.1 Smear Reduction	82
A FlyCapture API Code Samples	83
A.1 Setting a GPIO Pin to Strobe Using the FlyCapture API	83

A.2 Setting a Standard Video Mode, Format and Frame Rate Using the FlyCapture API	83
A.3 Asynchronous Hardware Triggering Using the FlyCapture API	83
A.4 Setting Brightness Using the FlyCapture API	84
A.5 Setting Shutter Using the FlyCapture API	84
A.6 Setting Gain Using the FlyCapture API	85
A.7 Setting Auto Exposure Using the FlyCapture API	85
A.8 Setting Sharpness Using the FlyCapture API	86
A.9 Setting Gamma Using the FlyCapture API	87
A.10 Setting White Balance Using the FlyCapture API	87
A.11 Accessing Raw Bayer Data using FlyCapture	88
A.12 Setting Hue Using the FlyCapture API	88
A.13 Setting Saturation Using the FlyCapture API	89
B FlyCapture SDK Examples	90
B.1 AsyncTriggerEx	90
B.2 BusEventsEx	90
B.3 CustomImageEx	90
B.4 ExtendedShutterEx	91
B.5 FlyCap2CameraControl	91
B.6 FlyCap2_GTKmm	91
B.7 FlyCap2MFC	92
B.8 FlyCapture2GUI	92
B.9 FlyCapture2SimpleGUI_WPF	92
B.10 FlyCapture2Test	93
B.11 GigEGrabEx	93
B.12 GrabCallbackEx	93
B.13 HighDynamicRangeEx	93
B.14 ImageEventEx	94
B.15 MultipleCameraEx	95
B.16 MultipleCameraWriteToDiskEx	95
B.17 MultiSyncEx	95
B.18 SaveImageToAviEx	96
B.19 SaveImageToFlashEx	96
B.20 SerialPortEx	96
C GenICam Features	98

C.1 Device Control	98
C.2 Analog Control	100
C.3 Image Format Control	101
C.4 Acquisition Control	102
C.5 Digital Input Output Control	103
C.6 Transport Layer Control	104
C.7 User Set Control	105
C.8 Chunk Data Control	105
C.9 LUT Control	106
D Control and Status Registers	107
D.1 IMAGE_RETRANSMIT: 634h	107
D.2 DATA_FLASH_CTRL: 1240h	108
D.3 DATA_FLASH_DATA: 1244h	108
D.4 GPIO_XTRA: 1104h	108
D.5 OUTPUT_VOLTAGE_ENABLE: 19D0h	109
D.6 AE_ROI: 1A70 – 1A74h	109
D.7 FRAME_INFO: 12F8h	110
D.8 INITIALIZE: 000h	111
D.9 TEMPERATURE: 82Ch	111
D.10 VMODE_ERROR_STATUS: 628h	111
D.11 CAMERA_LOG: 1D00 – 1DFFh	112
D.12 LED_CTRL: 1A14h	112
D.13 PIXEL_DEFECT_CTRL: 1A60h	112

About This Manual

This manual provides the user with a detailed specification of the Blackfly USB 3.1 camera system. The user should be aware that the camera system is complex and dynamic – if any errors or omissions are found during experimentation, please contact us.

This document is subject to change without notice.

Note: All model-specific information presented in this manual reflects functionality available in the model's firmware version. For more information see [Camera Firmware](#).

Where to Find Information

Chapter	What You Will Find
Blackfly USB 3.1 Specifications	General camera specifications and specific model specifications, and camera properties.
Blackfly USB 3.1 Installation	Instructions for installing the Blackfly USB 3.1, as well as introduction to Blackfly USB 3.1 configuration.
Tools to Control the Blackfly USB 3.1	Information on the tools available for controlling the Blackfly USB 3.1.
Blackfly USB 3.1 Physical Interface	Information on the mechanical properties of the Blackfly USB 3.1.
General Blackfly USB 3.1 Operation	Information on powering the Blackfly USB 3.1, monitoring status, user configuration sets, memory controls, and firmware.
Input/Output Control	Information on input/output modes and controls.
Image Acquisition	Information on asynchronous triggering and supported trigger modes.
Blackfly USB 3.1 Attributes	Information on supported imaging parameters and their controls.
Troubleshooting	Information on how to get support, diagnostics for the Blackfly USB 3.1, and common sensor artifacts.
Appendix: FlyCapture API Code Samples	Examples of FlyCapture API code.
Appendix: FlyCapture SDK Examples	Sample programs provided with the FlyCapture SDK.
Appendix: GenICam Features	Information on GenICam Feature controls.
Appendix: Control and Status Registers	Information on IIDC Control and Status Registers for functions not handled via GenICam.

Document Conventions

This manual uses the following to provide you with additional information:

Note: A note that contains information that is distinct from the main body of text. For example, drawing attention to a difference between models; or a reminder of a limitation.

Warning! A note that contains a warning to proceed with caution and care, or to indicate that the information is meant for an advanced user. For example, indicating that an action may void the camera's warranty or cause damage to the camera or other equipment.

Code is presented in a grey box with Courier font.

If further information can be found in our Knowledge Base, a list of articles is provided.

Related Knowledge Base Articles

Title	Article
Title of the Article	Link to the article on the website

If there are further resources available, a link is provided either to an external website, or to the SDK.

Related Resources

Title	Link
Title of the resource	Link to the resource

Contacting Us

For any questions, concerns or comments please contact us via the following methods:

Email	General questions Technical support (existing customers only)
Knowledge Base	Find answers to commonly asked questions in our Knowledge Base
Downloads	Download the latest documents and software
Contact Information	Contact Us on our website

Revision History

Revision	Date	Notes
1.0	June 12, 2014	Support for BFLY-U3-13S2
1.1	July 16, 2014	Clarified Input/Output section Added Video Mode descriptions
2.0	September 8, 2014	Support for BFLY-U3-03S2
3.0	December 2, 2014	Support for BFLY-U3-05S2
4.0	January 28, 2015	Support for BFLY-U3-20S4
5.0	June 2, 2015	Support for BFLY-U3-23S6 Support for BFLY-U3-50S5 Added firmware version to specifications tables
6.0	January 30, 2017	New template Added GPIO operating range For BFLY-U3-23S6C: added HDR; added Mode 1 in Video Mode Document exposure compensation in GenICam Minor bug fixes
6.1	March 15, 2017	Added C-mount material
6.2	August 30, 2018	Updated barrel lengths to one significant digit. Updated GPIO electrical input/output timing tables

1 Blackfly USB 3.1 Specifications

1.1 BFLY-U3-03S2 Specifications

	BFLY-U3-03S2M-CS	BFLY-U3-03S2C-CS
Firmware	1.8.3.0	1.8.3.0
Resolution	648 x 488	648 x 488
Frame Rate	84 FPS	84 FPS
Megapixels	0.3 MP	0.3 MP
Chroma	Mono	Color
Sensor	Sony ICX424, CCD, 1/3"	Sony ICX424, CCD, 1/3"
Readout Method	Global shutter	Global shutter
Pixel Size	7.4 μ m	7.4 μ m
Lens Mount	CS-mount	CS-mount
ADC	12-bit	12-bit
Gain Range	-10.104 dB to 23.991 dB	-10.104 dB to 23.991 dB
Exposure Range	0.023 ms to 31.98 seconds	0.023 ms to 31.98 seconds
Trigger Modes	Standard, bulb, low smear, overlapped, multi-shot	Standard, bulb, low smear, overlapped, multi-shot
Partial Image Modes	Pixel binning, decimation, ROI	Pixel binning, decimation, ROI
Image Processing	Gamma, lookup table, and sharpness	Gamma, lookup table, hue, saturation, and sharpness
Image Buffer	16 MB	16 MB
User Sets	2 memory channels for custom camera settings	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output	1 input, 1 output
Auxiliary Output	3.3 V, 120 mA maximum	3.3 V, 120 mA maximum
Interface	USB 3.1	USB 3.1
Power Requirements	5-24 V via GPIO or 5 V via USB 3.1 interface	5-24 V via GPIO or 5 V via USB 3.1 interface
Power Consumption	<3 W maximum	<3 W maximum
Dimensions/Mass	29 mm x 29 mm x 30 mm/36 g	29 mm x 29 mm x 30 mm/36 g
Machine Vision Standard	USB3 Vision v1.0	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature	Operating: 0° to 45°C Storage: -30° to 60°C	Operating: 0° to 45°C Storage: -30° to 60°C
Humidity	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)
Warranty	3 years	3 years

1.2 BFLY-U3-05S2 Specifications

	BFLY-U3-05S2M-CS	BFLY-U3-05S2C-CS
Firmware	1.4.3.1	1.4.3.1
Resolution	808 x 608	808 x 608
Frame Rate	50 FPS	50 FPS
Megapixels	0.5 MP	0.5 MP
Chroma	Mono	Color
Sensor	Sony ICX693, CCD, 1/3"	Sony ICX693, CCD, 1/3"
Readout Method	Global shutter	Global shutter
Pixel Size	6.0 µm	6.0 µm
Lens Mount	CS-mount	CS-mount
ADC	12-bit	12-bit
Gain Range	0 dB to 23.991 dB	0 dB to 23.991 dB
Exposure Range	0.031 ms to 31.9 seconds	0.031 ms to 31.9 seconds
Trigger Modes	Standard, bulb, low smear, overlapped, multi-shot	Standard, bulb, low smear, overlapped, multi-shot
Partial Image Modes	Pixel binning, decimation, ROI	Pixel binning, decimation, ROI
Image Processing	Gamma, lookup table, and sharpness	Gamma, lookup table, hue, saturation, and sharpness
Image Buffer	16 MB	16 MB
User Sets	2 memory channels for custom camera settings	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output	1 input, 1 output
Auxiliary Output	3.3 V, 120 mA maximum	3.3 V, 120 mA maximum
Interface	USB 3.1	USB 3.1
Power Requirements	5-24 V via GPIO or 5 V via USB 3.1 interface	5-24 V via GPIO or 5 V via USB 3.1 interface
Power Consumption	<3 W maximum	<3 W maximum
Dimensions/Mass	29 mm x 29 mm x 30 mm/36 g	29 mm x 29 mm x 30 mm/36 g
Machine Vision Standard	USB3 Vision v1.0	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature	Operating: 0° to 45°C Storage: -30° to 60°C	Operating: 0° to 45°C Storage: -30° to 60°C
Humidity	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)
Warranty	3 years	3 years

1.3 BFLY-U3-13S2 Specifications

	BFLY-U3-13S2M-CS	BFLY-U3-13S2C-CS
Firmware	1.1.3.0	1.1.3.0
Resolution	1288 x 964	1288 x 964
Frame Rate	30 FPS	30 FPS
Megapixels	1.3 MP	1.3 MP
Chroma	Mono	Color
Sensor	Sony ICX445, CCD, 1/3"	Sony ICX445, CCD, 1/3"
Readout Method	Global shutter	Global shutter
Pixel Size	3.75 µm	3.75 µm
Lens Mount	CS-mount	CS-mount
ADC	12-bit	12-bit
Gain Range	-11 dB to 23.99 dB	-11 dB to 23.99 dB
Exposure Range	0.034 ms to 31.9 seconds	0.034 ms to 31.9 seconds
Trigger Modes	Standard, bulb, low smear, overlapped, multi-shot	Standard, bulb, low smear, overlapped, multi-shot
Partial Image Modes	Pixel binning, decimation, ROI	Pixel binning, decimation, ROI
Image Processing	Gamma, lookup table, and sharpness	Gamma, lookup table, hue, saturation, and sharpness
Image Buffer	16 MB	16 MB
User Sets	2 memory channels for custom camera settings	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output	1 input, 1 output
Auxiliary Output	3.3 V, 120 mA maximum	3.3 V, 120 mA maximum
Interface	USB 3.1	USB 3.1
Power Requirements	5-24 V via GPIO or 5 V via USB 3.1 interface	5-24 V via GPIO or 5 V via USB 3.1 interface
Power Consumption	<3 W maximum	<3 W maximum
Dimensions/Mass	29 mm x 29 mm x 30 mm/36 g	29 mm x 29 mm x 30 mm/36 g
Machine Vision Standard	USB3 Vision v1.0	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature	Operating: 0° to 45°C Storage: -30° to 60°C	Operating: 0° to 45°C Storage: -30° to 60°C
Humidity	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)
Warranty	3 years	3 years

1.4 BFLY-U3-20S4 Specifications

	BFLY-U3-20S4M-CS	BFLY-U3-20S4C-CS
Firmware	1.6.3.0	1.6.3.0
Resolution	1624 x 1224	1624 x 1224
Frame Rate	15 FPS	15 FPS
Megapixels	2.0 MP	2.0 MP
Chroma	Mono	Color
Sensor	Sony ICX274, CCD, 1/1.8"	Sony ICX274, CCD, 1/1.8"
Readout Method	Global shutter	Global shutter
Pixel Size	4.4 µm	4.4 µm
Lens Mount	CS-mount	CS-mount
ADC	12-bit	12-bit
Gain Range	-10.319 dB to 23.991 dB	-10.319 dB to 23.991 dB
Exposure Range	0.072 ms to 31.9 seconds	0.072 ms to 31.9 seconds
Trigger Modes	Standard, bulb, low smear, overlapped, multi-shot	Standard, bulb, low smear, overlapped, multi-shot
Partial Image Modes	Pixel binning, decimation, ROI	Pixel binning, decimation, ROI
Image Processing	Gamma, lookup table, and sharpness	Gamma, lookup table, hue, saturation, and sharpness
Image Buffer	16 MB	16 MB
User Sets	2 memory channels for custom camera settings	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output	1 input, 1 output
Auxiliary Output	3.3 V, 120 mA maximum	3.3 V, 120 mA maximum
Interface	USB 3.1	USB 3.1
Power Requirements	5-24 V via GPIO or 5 V via USB 3.1 interface	5-24 V via GPIO or 5 V via USB 3.1 interface
Power Consumption	<3 W maximum	<3 W maximum
Dimensions/Mass	29 mm x 29 mm x 30 mm/36 g	29 mm x 29 mm x 30 mm/36 g
Machine Vision Standard	USB3 Vision v1.0	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature	Operating: 0° to 45°C Storage: -30° to 60°C	Operating: 0° to 45°C Storage: -30° to 60°C
Humidity	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)
Warranty	3 years	3 years

1.5 BFLY-U3-23S6 Specifications

	BFLY-U3-23S6M-C	BFLY-U3-23S6C-C
Firmware	1.8.3.0	1.8.3.0
Resolution	1920 x 1200	1920 x 1200
Frame Rate	41 FPS	41 FPS
Megapixels	2.3 MP	2.3 MP
Chroma	Mono	Color
Sensor	Sony IMX249, CMOS, 1/1.2"	Sony IMX249, CMOS, 1/1.2"
Readout Method	Global shutter	Global shutter
Pixel Size	5.86 µm	5.86 µm
Lens Mount	C-mount	C-mount
ADC	10-bit / 12-bit in Mode 7	10-bit / 12-bit in Mode 7
Gain Range	0 dB to 29.996 dB	0 dB to 29.996 dB
Exposure Range	0.019 ms to 3.9 seconds	0.019 ms to 3.9 seconds
Trigger Modes	Standard, bulb, overlapped, multi-shot	Standard, bulb, overlapped, multi-shot
Partial Image Modes	Pixel binning, decimation, ROI	Pixel binning, decimation, ROI
Image Processing	Gamma, lookup table, and sharpness	Gamma, lookup table, hue, saturation, and sharpness
Image Buffer	16 MB	16 MB
User Sets	2 memory channels for custom camera settings	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output	1 input, 1 output
Auxiliary Output	3.3 V, 120 mA maximum	3.3 V, 120 mA maximum
Interface	USB 3.1	USB 3.1
Power Requirements	5-24 V via GPIO or 5 V via USB 3.1 interface	5-24 V via GPIO or 5 V via USB 3.1 interface
Power Consumption	<3 W maximum	<3 W maximum
Dimensions/Mass	29 mm x 29 mm x 30 mm/36 g	29 mm x 29 mm x 30 mm/36 g
Machine Vision Standard	USB3 Vision v1.0	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature	Operating: 0° to 45°C Storage: -30° to 60°C	Operating: 0° to 45°C Storage: -30° to 60°C
Humidity	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)
Warranty	3 years	3 years

1.6 BFLY-U3-50H5 Specifications

	BFLY-U3-50H5M-C	BFLY-U3-50H5C-C
Firmware	1.8.3.0	1.8.3.0
Resolution	2448 x 2048	2448 x 2048
Frame Rate	7.5 FPS	7.5 FPS
Megapixels	5.0 MP	5.0 MP
Chroma	Mono	Color
Sensor	Sharp RJ32S4AA0DT, CCD, 2/3"	Sharp RJ32S3AA0DT, CCD, 2/3"
Readout Method	Global shutter	Global shutter
Pixel Size	3.45 µm	3.45 µm
Lens Mount	C-mount	C-mount
ADC	12-bit	12-bit
Gain Range	-6.234 dB to 23.991 dB	-6.234 dB to 23.991 dB
Exposure Range	0.082 ms to 32 seconds	0.082 ms to 32 seconds
Trigger Modes	Standard, bulb, low smear, overlapped, multi-shot	Standard, bulb, low smear, overlapped, multi-shot
Partial Image Modes	Pixel binning, decimation, ROI	Pixel binning, decimation, ROI
Image Processing	Gamma, lookup table, and sharpness	Gamma, lookup table, hue, saturation, and sharpness
Image Buffer	16 MB	16 MB
User Sets	2 memory channels for custom camera settings	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output	1 input, 1 output
Auxiliary Output	3.3 V, 120 mA maximum	3.3 V, 120 mA maximum
Interface	USB 3.1	USB 3.1
Power Requirements	5-24 V via GPIO or 5 V via USB 3.1 interface	5-24 V via GPIO or 5 V via USB 3.1 interface
Power Consumption	<3 W maximum	<3 W maximum
Dimensions/Mass	29 mm x 29 mm x 30 mm/36 g	29 mm x 29 mm x 30 mm/36 g
Machine Vision Standard	USB3 Vision v1.0	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature	Operating: 0° to 45°C Storage: -30° to 60°C	Operating: 0° to 45°C Storage: -30° to 60°C
Humidity	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)	Operating: 20% to 80% (no condensation) Storage: 20% to 95% (no condensation)
Warranty	3 years	3 years

1.7 Analog-to-Digital Converter (ADC)

All CCD camera sensors incorporate an ADC to digitize the images.

All CMOS camera sensors incorporate an on-chip ADC to digitize the images.

The Blackfly USB 3.1's ADC is configured to a fixed bit output. If the [pixel format](#) selected has fewer bits per pixel than the ADC output, the least significant bits are dropped. If the pixel format selected has greater bits per pixel than the ADC output, the least significant bits are padded and can be discarded by the user. Image data is left-aligned across a 2-byte format.

For example, for a 12-bit output, the least significant 4 bits are padded in order to fill 2 bytes. E.g. 0xFFFF0.

A 10-bit conversion produces 1,024 possible values between 0 and 65,472.

A 12-bit conversion produces 4,096 possible values between 0 and 65,520.

The following table illustrates the most important aspects of the ADC.

Resolution	12-bit, 50 MHz
Black Level Clamp	0 LSB to 255.75 LSB, 0.25 LSB steps
Pixel Gain Amplifier	-3 dB to 6 dB, 3 dB steps
Variable Gain Amplifier	6 dB to 42 dB, 10-bit

The bit depth of the output varies between sensors and can be seen in the table below.

Model	ADC
BFLY-U3-03S2M-CS	12-bit
BFLY-U3-03S2C-CS	12-bit
BFLY-U3-05S2M-CS	12-bit
BFLY-U3-05S2C-CS	12-bit
BFLY-U3-13S2M-CS	12-bit
BFLY-U3-13S2C-CS	12-bit
BFLY-U3-20S4M-CS	12-bit
BFLY-U3-20S4C-CS	12-bit
BFLY-U3-23S6M-C	10-bit (12-bit in Mode 7)
BFLY-U3-23S6C-C	10-bit (12-bit in Mode 7)
BFLY-U3-50H5M-C	12-bit
BFLY-U3-50H5C-C	12-bit

2 Blackfly USB 3.1 Installation

2.1 Before You Install

2.1.1 Will your system configuration support the camera?

Recommended System Configuration

Operating System	CPU	RAM	Video	Ports	Software
Windows, Linux (32- and 64-bit)	Intel Core 2 Duo, or equivalent	2 GB RAM	128 MB	PCIe 2.0 compatible host controller with USB 3.1 connector	Microsoft Visual Studio 2010 (to compile and run example code)

Note: Refer to [Technical Application Note 10359](#) for important information on recommended and unsupported USB 3.1 system components.

2.1.2 Do you have all the parts you need?

To install your camera you will need the following components:

- USB 3.1 cable (see [Interface Cables](#))
- 6-pin GPIO cable (see [General Purpose Input/Output \(GPIO\)](#))
- Lens (see [Lens Mounting](#))
- Tripod adapter (optional) (see [Mounting with the Case or Mounting Bracket](#))
- Interface card (see [Interface Card](#))

FLIR sells a number of the additional parts required for installation. To purchase, visit our [Accessories page](#).

2.1.3 Do you have a downloads account?

Our [downloads](#) page has many resources to help you operate your camera effectively, including:

- Software, including Drivers (required for installation)
- Firmware updates and release notes
- Dimensional drawings and CAD models
- Documentation

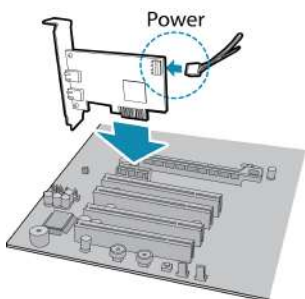
To access the downloads resources you must have a downloads account.

1. Go to our website: www.flir.com/iis.
2. In the upper right corner, click **Register**.
3. Complete the form, then click **Register**.

After you submit your registration, you will receive an email with instructions on how to activate your account.

2.2 Installing Your Interface Card and Software

1. Install your Interface Card



Ensure the card is installed per the manufacturer's instructions.

Connect the internal IDE or SATA power connector on the card to the computer power supply.

Alternatively, use your PC's built-in host controller, if equipped.

Open the Windows Device Manager. Ensure the card is properly installed under **Universal Serial Bus Controllers**. An exclamation point (!) next to the card indicates the driver has not yet been installed.

2. Install the FlyCapture® Software

Note: For existing users who already have FlyCapture installed, we recommend ensuring you have the latest version for optimal performance of your camera. If you do not need to install FlyCapture, use the DriverControlGUI to install and enable drivers for your card.

- a. Login to our [downloads](#) page.
- b. Select your **Camera** and **Operating System** from the drop-down lists and click the **Search** button.
- c. Click on the **Software** search results to expand the list.
- d. Click the appropriate link to begin the download and installation.

After the download is complete, the FlyCapture setup wizard begins. If the wizard does not start automatically, double-click the .exe file to open it. Follow the steps in each setup dialog.

3. Enable the Drivers for the card

During installation, you are prompted to select your interface driver.

In the **Interface Driver Selection** dialog, select the **I will use USB cameras**.

This selection ensures the pgrxhci (USBPro) and pgrusbcam drivers are installed.

To uninstall or reconfigure the driver at any time after setup is complete, use the DriverControlGUI (see [Configuring Camera Setup](#)).

2.3 Installing Your Camera

1. Install the Tripod Mounting Bracket (optional)

The ASA and ISO-compliant tripod mounting bracket attaches to the camera using the included metal screws.

2. Attach a Lens

Unscrew the dust cap from the CS-mount lens holder to install a lens. Note: the camera can be used with a removable 5 mm C- mount adapter.

Unscrew the dust cap from the C-mount lens holder to install a lens.

3. Connect the interface Card and Cable to the Camera



Plug the interface cable into the host controller card and the camera. The cable jack screws can be used for a secure connection.

4. Plug in the GPIO connector

5. Confirm Successful Installation

Check Device Manager to confirm that installation was successful.

- a. Go to the **Start menu**, select **Run**, and enter **devmgmt.msc**.

Verify the camera is listed under "**Point Grey Research Devices**."

- b. Run the FlyCap program: **All Programs->Point Grey FlyCapture2 SDK-> FlyCap**

The FlyCap program can be used to test the camera's image acquisition capabilities.

Changes to your camera's installation configuration can be made using utilities available in the FlyCapture SDK (see [Configuring Camera Setup](#)).

2.4 Configuring Camera Setup

After successful installation of your camera and interface card, you can make changes to the setup. Use the tools described below to change the driver for your interface card.

For information on updating your camera's firmware post installation, see [Camera Firmware](#).

2.4.1 Configuring Camera Drivers

A FlyCapture SDK installation comes with two USB 3.1 drivers: xhci (USBPro) and pgrusbcam.

The PGRxHCI driver does not support USB devices from other manufacturers.

Related Knowledge Base Articles

Title	Article
Recommended USB 3.1 System Components	Technical Application Note 10359
How does my USB 3.1 camera appear in Device Manager?	Knowledge Base Article 10107

To manage and update drivers use the DriverControlGUI utility provided in the SDK. To open the DriverControlGUI:

Start Menu-->All Programs-->FlyCapture SDK-->Utilities-->DriverControlGUI

Select the interface from the tabs in the top left. Then select your interface card to see the current setup.

For more information about using the DriverControlGUI, see the online help provided in the tool.

2.4.2 Maximum Number of Cameras on a Single Bus

A single USB port generally constitutes a single 'bus.' The USB 3.1 standard allows for multiple devices to be connected to a single bus. The number of cameras is limited by the following considerations:

- Adequate power supply. The camera requires a nominal 5 V to operate effectively. While a standard, non-powered bus provides 500 mA of current at 5 V, an internal, bus-powered hub provides only 400 mA. Externally-powered hubs provide 500 mA per port.
- Adequate bandwidth. The effective bandwidth available via the USB 3.1 bulk transfer method is 384 MB per second. A PCIe 2.0 interface or greater is required to use the full bandwidth. PCIe 2.0 can transfer just under 400 MB per second. However, bandwidth must be shared on the system, depending on the operating configuration of the cameras (resolution, frame rate, and pixel format).

Related Knowledge Base Articles

Title	Article
Setting up multiple USB 3.1 cameras	Technical Application Note 10350

3 Tools to Control the Blackfly USB 3.1

The Blackfly USB 3.1's features can be accessed using various controls, including:

- Spinnaker SDK including API examples and the SpinView program
- FlyCapture SDK including API examples and the FlyCap program
- GenICam Applications
- Control and Status Registers
- Third-party Software Applications

Examples of the controls are provided throughout this document. Additional information can be found in the appendices.

3.1 Using Spinnaker®

The user can monitor or control features of the camera through Spinnaker API examples provided in the Spinnaker SDK, or through the SpinView Program.

3.1.1 SpinView Program

The SpinView application is a generic, easy-to-use streaming image viewer included with the Spinnaker SDK that can be used to test many of the capabilities of your camera. It allows you to view a live video stream from the camera, save individual images, adjust the various attributes, frame rates, features and settings of the camera. It includes tools for updating firmware, managing drivers, IP addressing, and activity logging.

3.1.2 Custom Applications Built with the Spinnaker API

The Spinnaker SDK includes a full Application Programming Interface that allows you to create custom applications to control your camera. Included with the SDK are a number of source code examples to help you get started.

Spinnaker API examples are provided for C, C++, C#, and VB.NET languages. These examples are precompiled for your convenience.

3.2 Using FlyCapture®

The user can monitor or control features of the camera through FlyCapture API examples provided in the FlyCapture SDK, or through the FlyCap Program.

3.2.1 FlyCap Program

The FlyCap application is a generic, easy-to-use streaming image viewer included with the FlyCapture SDK that can be used to test many of the capabilities of your compatible camera. It allows you to view a live video stream from the camera, save individual images, adjust the various video formats, frame rates, properties and settings of the camera, and access camera registers directly. Consult the FlyCapture SDK Help for more information.

3.2.2 Custom Applications Built with the FlyCapture API

The FlyCapture SDK includes a full Application Programming Interface that allows customers to create custom applications to control cameras. Included with the SDK are a number of source code examples to help programmers get started.

FlyCapture API examples are provided for C, C++, C#, and VB.NET languages. There are also a number of precompiled examples.

Code samples are provided in [FlyCapture API Code Samples](#).

Examples of basic programming tasks are described in [FlyCapture SDK Examples](#)

3.3 Using GenICam Applications

USB3 Vision is a communication interface for vision applications based on the USB 3.1 technology. All cameras supporting USB3 Vision interact the same way with software also supporting USB3 Vision.

The standard defines required elements for camera identification, control, and output. It uses GenICam, a programming interface for camera attribute control. GenICam allows camera vendors to define features and attributes in an XML file stored inside the camera. The file is parsed by the host application when the camera is initially discovered. One of the key benefits of GenICam is the ability for camera vendors to introduce new camera-specific features without needing to update the host application.

Each camera attribute, such as exposure time, is controlled by a specific GenICam feature. The camera includes an XML device description file for interfacing with third-party GenICam-compliant APIs. A full listing of features that are included in the XML file is provided in [GenICam Features](#).

Not all operations can be controlled using the XML file; those not included are controlled via Control and Status Registers (CSRs). These registers conform to the IIDC v1.32 standard. A complete list of CSRs can be found in the *FLIR Machine Vision Camera Register Reference* available from the [Downloads](#) page.

Throughout this document, GenICam features are referenced with their applicable operation; where no GenICam feature is available in the XML file, the CSR is referenced.

For more information on the USB3 Vision standard, visit visiononline.org.

For more information on GenICam, visit emva.org.

Related Knowledge Base Articles

Title	Article
Using USB3 Vision cameras with National Instruments' Acquisition Software	Technical Application Note 10337
Using USB3 Vision cameras with A&B Software's ActiveUSB	Technical Application Note 10335
Using USB3 Vision cameras with Matrox Imaging Library	Technical Application Note 10701
Using USB3 Vision cameras with MVTec's Halcon software	Technical Application Note 10774
Using USB3/USB2 cameras with Cognex VisionPro	Technical Application Note 10788

3.4 Using Control and Status Registers

The user can monitor or control each feature of the camera through the control and status registers (CSRs) programmed into the camera firmware. These registers conform to the IIDC v1.32 standard (except where noted). *Format* tables for

each 32-bit register are presented to describe the purpose of each bit that comprises the register. Bit 0 is always the most significant bit of the register value.

Register offsets and values are generally referred to in their hexadecimal forms, represented by either a '0x' before the number or 'h' after the number, e.g. the decimal number 255 can be represented as 0xFF or FFh.

A complete list of CSRs can be found in the [FLIR Machine Vision Camera Register Reference](#).

The controllable fields of most registers are *Mode* and *Value*.

Modes

Each CSR has three bits for mode control, ON_OFF, One_Push and A_M_Mode (Auto/Manual mode). Each feature can have four states corresponding to the combination of mode control bits.

Note: Not all features implement all modes.

Table 3.1: CSR Mode Control Descriptions

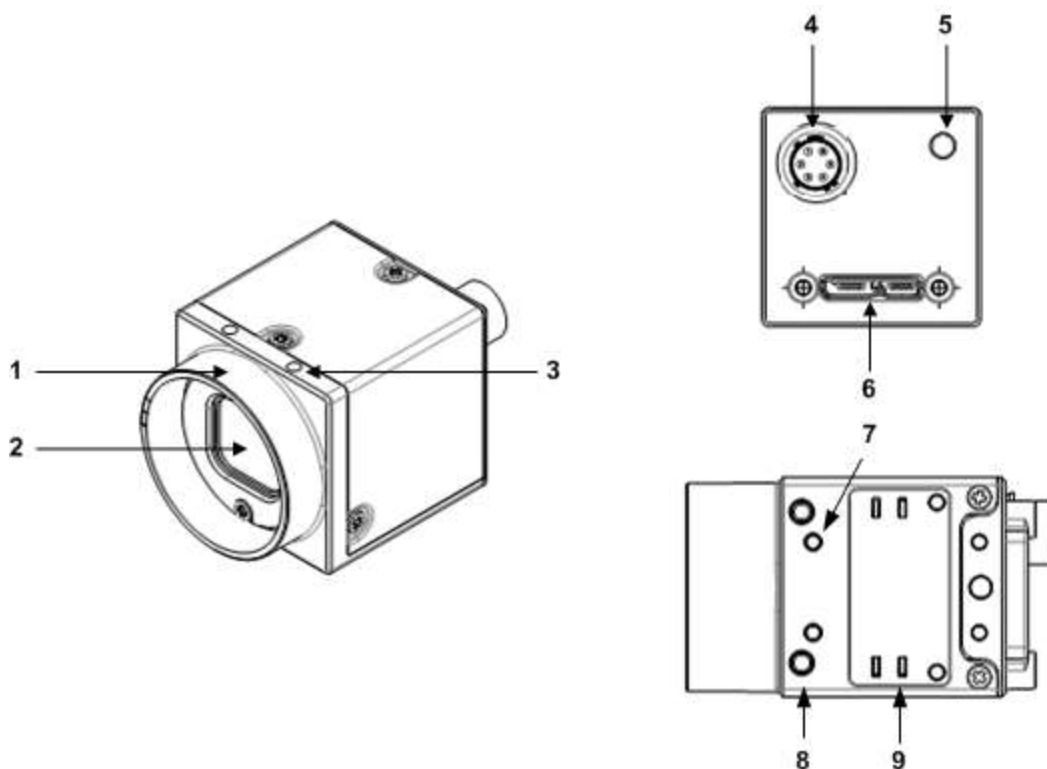
One_Push	ON_OFF	A_M_Mode	State
N/A	0	N/A	Off state. Feature will be fixed value state and uncontrollable.
N/A	1	1	Auto control state. Camera controls feature by itself continuously.
0	1	0	Manual control state. User can control feature by writing value to the value field.
1 (Self clear)	1	0	One-Push action. Camera controls feature by itself only once and returns to the Manual control state with adjusted value.

Values

If the *Presence_Inq* bit of the register is one, the *value* field is valid and can be used for controlling the feature. The user can write control values to the *value* field only in the **Manual control state**. In the other states, the user can only read the *value*. The camera always has to show the real setting value at the *value* field if *Presence_Inq* is one.

4 Blackfly USB 3.1 Physical Interface

4.1 Blackfly USB 3.1 Physical Description



1. Lens holder

See [Lens Mounting](#)

2. Glass/IR filter system

See [Dust Protection](#) and [Infrared Cut-Off Filters](#)

3. M2x2.5 mounting holes

See [Mounting with the Case or Mounting Bracket](#)

4. General purpose I/O connector

See [General Purpose Input/Output \(GPIO\)](#)

5. Status LED

See [Status Indicator LED](#)

6. USB3 connector

See [USB 3.1 Connector](#)

7. M2x2.5 mounting holes

8. M3x2.5 mounting holes

See [Mounting with the Case or Mounting Bracket](#)

9. Camera label

Contains camera information such as model name, serial number and required compliance.

4.2 Blackfly USB 3.1 Dimensions

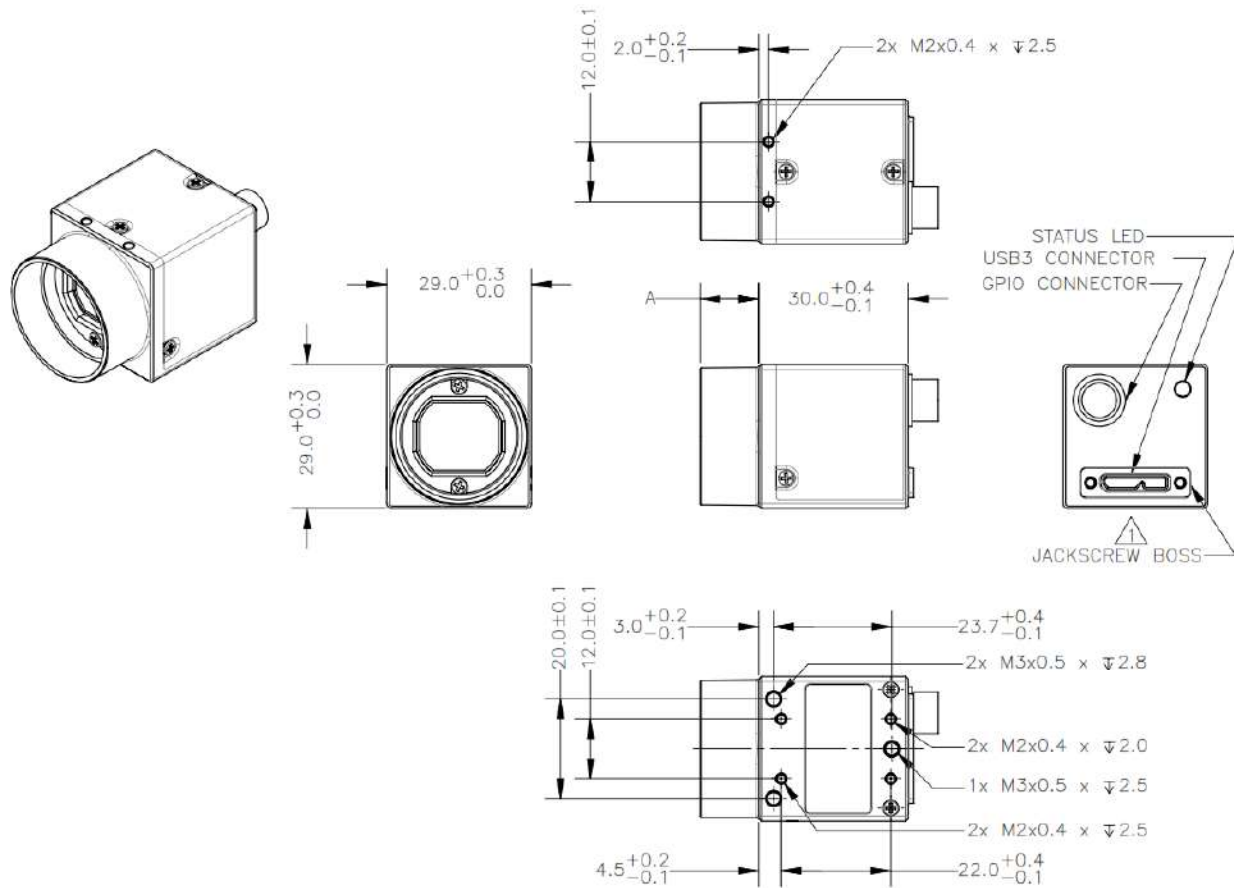


Figure 4.1: Blackfly USB 3.1 Dimensional Drawing

Model	Barrel Length "A"
BFLY-U3-03S2	7.3
BFLY-U3-05S2	7.3
BFLY-U3-13S2	7.3
BFLY-U3-20S4	7.3
BFLY-U3-23S6	11.8
BFLY-U3-50H5	12.3

Note: To obtain 3D models, go to our [Downloads](#) site or contact [Support](#).

4.3 Mounting with the Case or Mounting Bracket

Using the Case

The case is equipped with the following mounting holes:

- Two (2) M2 x 2mm mounting holes on the top of the case
- Three (3) M3 x 2.5mm mounting holes on the bottom of the case
- Four (4) M2 x 2mm mounting holes on the bottom of the case that can be used to attach the camera directly to a custom mount or to the tripod mounting bracket

Using the Mounting Bracket

The tripod mounting bracket is equipped with four (4) M2 mounting holes.

Tripod Adapter Dimensions

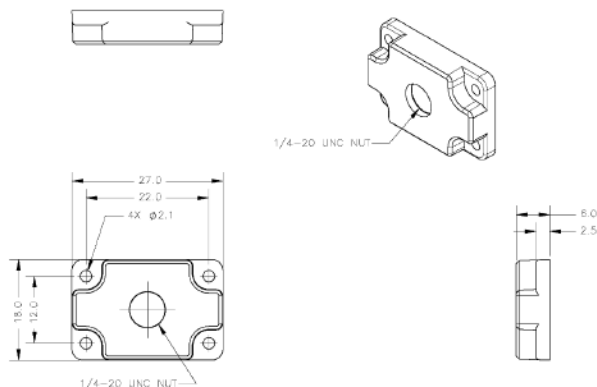


Figure 4.2: Tripod Adapter Dimensional Diagram

4.4 Lens Mounting

Lenses are not included with individual cameras.

Related Knowledge Base Articles

Title	Article
Selecting a lens for your camera	Knowledge Base Article 10694

The lens mount is compatible with CS-mount lenses.

The BFLY-U3-23S6 and BFLY-U3-50H5 lens mount is compatible with C-mount lenses. Correct focus cannot be achieved using a CS-mount lens on a C-mount camera.

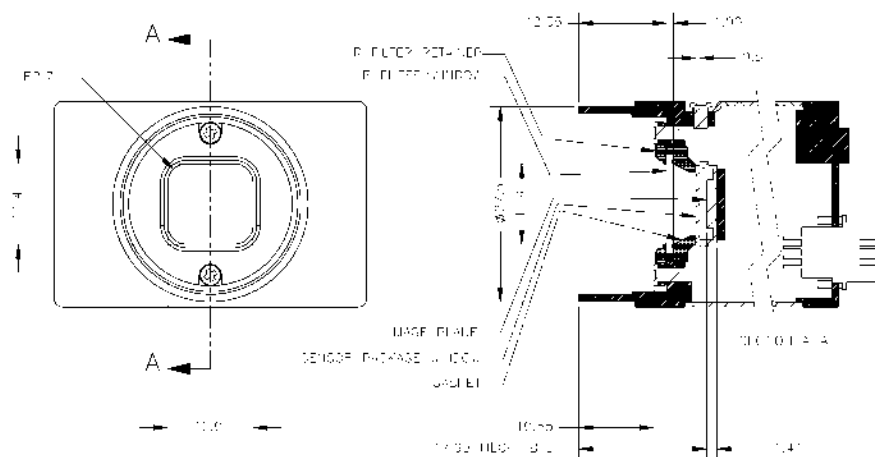


Figure 4.3: Example C-mount Cross Section

4.4.1 Back Flange Distance

The Back Flange Distance (BFD) is offset due to the presence of both a 1 mm infrared cutoff (IRC) filter and a 0.5 mm sensor package window. These two pieces of glass fit between the lens and the sensor image plane. The IRC filter is installed on color cameras. In monochrome cameras, it is a transparent piece of glass. The sensor package window is installed by the sensor manufacturer. Both components cause refraction, which requires some offset in flange back distance to correct.

The resulting C-mount BFD is 17.99 mm.

The resulting CS-mount BFD is 12.52 mm.

For more information about the IRC filter, see [Infrared Cut-Off Filters](#).

4.5 Handling Precautions and Camera Care

Warning! Do not open the camera housing. Doing so voids the Hardware Warranty described at the beginning of this manual.

Your camera is a precisely manufactured device and should be handled with care. Here are some tips on how to care for the device.

- Avoid electrostatic charging.
- When handling the camera unit, avoid touching the lenses. Fingerprints affect the quality of the image produced by the device.
- To clean the lenses, use a standard camera lens cleaning kit or a clean dry cotton cloth. Do not apply excessive force.
- Extended exposure to bright sunlight, rain, dusty environments, etc. may cause problems with the electronics and the optics of the system.
- Avoid excessive shaking, dropping or any kind of mishandling of the device.

Related Knowledge Base Articles

Title	Article
Cleaning the imaging surface of your camera	Knowledge Base Article 10243

4.5.1 Case Temperature and Heat Dissipation

You must provide sufficient heat dissipation to control the internal operating temperature of the camera.

The camera is equipped with an on-board temperature sensor. It allows you to obtain the temperature of the camera board-level components. The sensor measures the ambient temperature within the case.

Table 4.1: Temperature Sensor Specifications

Accuracy	±2.0°C from -25°C to +100°C ±3.0°C from -55°C to +120°C
Range	-55°C to +125°C
Resolution	0.125°C

Note: As a result of packing the camera electronics into a small space, the outer case of the camera can become very warm to the touch when running in some modes. This is expected behavior and will not damage the camera electronics.

To reduce heat, use a cooling fan to set up a positive air flow around the camera, taking into consideration the following precautions:

- Mount the camera on a heat sink, such as a camera mounting bracket, made out of a heat-conductive material like aluminum.
- Make sure the flow of heat from the camera case to the bracket is not blocked by a non-conductive material like plastic.
- Make sure the camera has enough open space around it to facilitate the free flow of air.

To access temperature information use:

- GenICam—[Device Control](#)

4.6 Dust Protection

The camera housing is designed to prevent dust from falling directly onto the sensor's protective glass surface. This is achieved by placing a piece of clear glass (monochrome camera models) or an IR cut-off filter (color models) that sits above the surface of the sensor's glass. A removable plastic retainer keeps this glass/filter system in place. By increasing the distance between the imaging surface and the location of the potential dust particles, the likelihood of interference from the dust (assuming non-collimated light) and the possibility of damage to the sensor during cleaning is reduced.

Warning!

- Cameras are sealed when they are shipped. To avoid contamination, seals should not be broken until cameras are ready for assembly at customer's site.
- Use caution when removing the protective glass or filter. Damage to any component of the optical path voids the Hardware Warranty.
- Removing the protective glass or filter alters the optical path of the camera, and may result in problems obtaining proper focus with your lens.

Related Knowledge Base Articles

Title	Article
Removing the IR filter from a color camera	Knowledge Base Article 10080
Selecting a lens for your camera	Technical Application Note 10694

4.7 Infrared Cut-Off Filters

FLIR machine vision color camera models are equipped with an additional infrared (IR) cut-off filter. This filter can reduce sensitivity in the near infrared spectrum and help prevent smearing. The properties of this filter are illustrated in the results below.

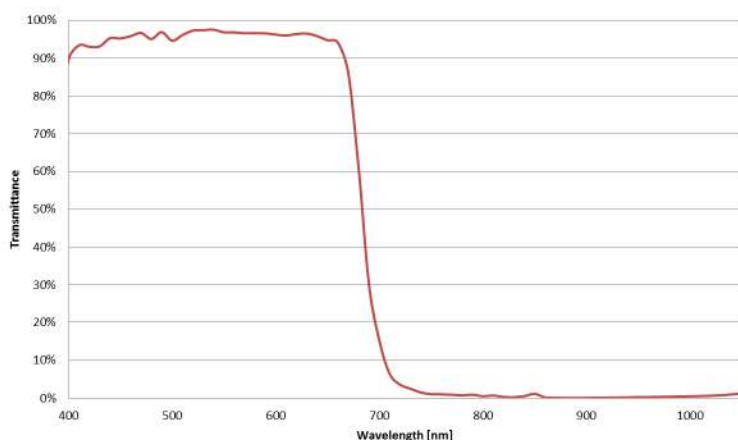


Figure 4.4: IR filter transmittance graph

In monochrome models, the IR filter is replaced with a transparent piece of glass.

The following are the properties of the IR filter/protective glass:

	1" and 1/1.2" Sensors	All Other Sensors
Type	Anti-reflective	Anti-reflective
Material	Schott B270	Schott B270
Dimensions	15.5 ±0.08 x 18 ±0.08 mm	14 ±0.08 x 14 ±0.08 mm
Thickness	1 ±0.07 mm	1 ±0.07 mm

For more information, see [Dust Protection](#).

Related Knowledge Base Articles

Title	Article
Removing the IR filter from a color camera	Knowledge Base Article 10080

4.8 Camera Interface and Connectors

4.8.1 USB 3.1 Connector

The camera is equipped with a USB 3.1 Micro-B connector that is used for data transmission, camera control and power. For more detailed information, consult the USB 3.1 specification available from <http://www.usb.org/developers/docs/>.



Figure 4.5: USB 3.1 Micro B Connector

Table 4.2: USB 3.1 Micro-B Connector Pin Assignments

Pin	Signal Name	Description
1	VBUS	Power
2	D-	USB 2.0 differential pair
3	D+	
4	ID	OTG identification
5	GND	Ground for power return
6	MicB_SSTX-	SuperSpeed transmitter differential pair
7	MicB_SSTX+	
8	GND_DRAIN	Ground for SuperSpeed signal return
9	MicB_SSRX-	SuperSpeed receiver differential pair
10	MicB_SSRX+	

The USB 3.1 Micro-B receptacle accepts a USB 2.0 Micro-B plug and, therefore, the camera is backward compatible with the USB 2.0 interface.

Note: When the camera is connected to a USB 2.0 interface, it runs at USB 2.0 speed, and maximum frame rates are adjusted accordingly based on current imaging parameters.

Related Knowledge Base Articles

Title	Article
USB 3.1 Frequently Asked Questions	Knowledge Base Article 10019

4.8.2 Interface Cables

The USB 3.1 standard does not specify a maximum cable length.

- 3-meter USB 3.1 cable ([ACC-01-2300](#))
- 5-meter USB 3.1 cable ([ACC-01-2301](#))

- 5-meter USB 3.1 cable high performance ([ACC-01-2302](#))

Note: A 5-meter cable (or longer) is not recommended for laptops or on board controllers.

To purchase a recommended cable from FLIR, visit our [Accessories](#) page.

4.8.3 Interface Card

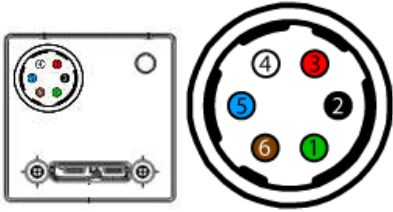
The camera must connect to an interface card. This is sometimes called a host adapter, a bus controller, or a network interface card (NIC).

In order to achieve the maximum benefits of USB 3.1, the camera must connect to a USB 3.1 PCIe 2.0 (or greater) card.

To purchase a compatible card from FLIR, visit our [Accessories](#) page.

4.8.4 General Purpose Input/Output (GPIO)

The camera is equipped with a 6-pin GPIO connector on the back of the case. The connector is a Hirose HR10A-7R-6PB, the mating connector is a Hirose HR10A-7P-6S(73).

Diagram	Color	Pin	Function	Description
	Green	1	V _{EXT}	+12 V DC Camera Power
	Black	2	I/O	Opto-isolated input (GPIO 0)
	Red	3	NC / +3.3 V	+3.3 V output. Current 120 mA (nominal). Firmware enabled (See OUTPUT_VOLTAGE_ENABLE: 19D0h)
	White	4	O1	Opto-isolated output (GPIO 1)
	Blue	5	OPTO_GND	Ground for opto-isolated I/O, not connected to camera ground
	Brown	6	GND	DC camera power ground

For more information on camera power, see [Powering the Camera](#).

For more information on configuring input/output with GPIO, see [Input/Output Control](#).

For details on GPIO circuits, see [GPIO Electrical Characteristics](#).

5 General Blackfly USB 3.1 Operation

5.1 Powering the Camera

The power consumption specification is: 5-24 V via GPIO or 5 V via USB 3.1 interface, maximum <3 W.

The USB 3.1 Micro-B connector provides a power connection between the camera and the host computer. The ideal input voltage is nominal 5 V DC.

Related Knowledge Base Articles

Title	Article
How can I power my USB 3.1 camera?	Knowledge Base Article 10240

Power can also be provided through the GPIO interface. For more information, see [Input/Output Control](#). The camera selects whichever power source is supplying a higher voltage.

Note: Some systems, especially those with laptop computers or longer cable lengths, may not provide adequate power through the USB 3.1 cable which could result in intermittent operation. The use of external power through the GPIO is required for these systems.

The camera does not transmit images for the first 100 ms after power-up. The auto-exposure and auto-white balance algorithms do not run while the camera is powered down. It may therefore take several (n) images to get a satisfactory image, where n is undefined.

When the camera is power cycled (power disengaged then re-engaged), the camera reverts to its default factory settings, or if applicable, the last saved memory channel. For more information, see [User Sets \(Memory Channels\)](#).

5.2 User Sets (Memory Channels)

The camera can save and restore settings and imaging parameters via on-board user configuration sets, also known as memory channels. This is useful for saving default power-up settings, such as gain, shutter, video format and frame rate, and others that are different from the factory defaults.

User Set 0 (or Memory channel 0) stores the factory default settings that can always be restored. Two additional user sets are provided for custom default settings. The camera initializes itself at power-up, or when explicitly reinitialized, using the contents of the last saved user set. Attempting to save user settings to the (read-only) factory default user set causes the camera to switch back to using the factory defaults during initialization.

The following camera settings are saved in user sets.

- Acquisition Frame Rate and Current Frame Rate
- Image Data Format, Position, and Size
- Image mirror, if applicable
- Current Video Mode and Current Video Format
- Camera power
- Frame information
- Trigger Mode and Trigger Delay
- Imaging Parameters such as: Brightness, Auto Exposure, Shutter, Gain, White Balance, Sharpness, Hue, Saturation, and Gamma
- Input/output controls such as: GPIO pin modes, GPIO strobe modes
- Color Coding ID/Pixel Coding

To access user sets:

- GenICam—[User Set Control](#)

5.2.1 GenICam User Set Control

Name	Display Name	Description	Value
CurrentUserSet	Current User Set	Indicates the user set that is currently in use. At initialization time, the camera loads the most recently saved user set	0 (default) 1 2
UserSetSelector	User Set Selector	Selects the user set to load or save	Default User Set 1 User Set 2
UserSetLoad	User Set Load	Loads the user set specified by the User Set Selector to the device and makes it active	Write Only
UserSetSave	User Set Save	Saves the user set specified by the User Set Selector to the non-volatile memory of the device	Write Only
DefaultUserSet	Default User Set	Selects the default user set as the default start up set	Default User Set 1 User Set 2

5.3 On-Camera Frame Buffer

The camera has a 16 MB that can be used for temporary image storage. This may be useful in cases such as:

- Retransmission of an image is required due to data loss or corruption.
- Multiple camera systems where there is insufficient bandwidth to capture images in the desired configuration.

All images pass through the frame buffer mechanism. This introduces relatively little delay in the system.

The frame buffer is volatile memory that is erased after power cycling. To store images on the camera after power cycling, use [Non-Volatile Flash Memory](#). Accessing flash memory is significantly slower than accessing the frame buffer, and storage is limited.

To control frame buffer:

- CSRs—[IMAGE_RETRANSMIT: 634h](#)

5.4 Non-Volatile Flash Memory

The camera has 1 MB non-volatile memory for users to store data.

- FlyCapture SDK example program—[SaveImageToFlashEx](#)

Related Knowledge Base Articles

Title	Article
Storing data in on-camera flash memory	Knowledge Base Article 10370

5.5 Camera Firmware

Firmware is programming that is inserted into the programmable read-only memory (programmable ROM) of most cameras. Firmware is created and tested like software. When ready, it can be distributed like other software and installed in the programmable read-only memory by the user.

The latest firmware versions often include significant bug fixes and feature enhancements. To determine the changes made in a specific firmware version, consult the Release Notes.

Firmware is identified by a version number, a build date, and a description.

Related Knowledge Base Articles

Title	Article
FLIR machine vision software and firmware version numbering scheme	Knowledge Base Article 10310
Determining the firmware version used by my camera	Knowledge Base Article 10312
Should I upgrade my camera firmware or software?	Knowledge Base Article 10216

5.5.1 Determining Firmware Version

To determine the firmware version number of your camera:

- In FlyCapture, open the Camera Control dialog and click on Camera Information.
- If you're implementing your own code, use `flycaptureGetCameraRegister()`.
- Query the GenICam Device Control feature `DeviceFirmwareVersion`.

5.5.2 Upgrading Camera Firmware

Camera firmware can be upgraded or downgraded to later or earlier versions using the UpdatorGUI program that is bundled with the FlyCapture SDK available from our [downloads site](#).

Before upgrading firmware:

- Ensure that FlyCapture2.dll is installed in the same directory as UpdatorGUI3.
- Download the firmware file from our [downloads site](#).

Warning! Do not disconnect the camera during the firmware update process.

To upgrade the firmware using Spinnaker:

1. **Start Menu-->All Programs-->Point Grey Spinnaker SDK-->SpinView**
2. From the Device list, right click the camera and select Update Device Firmware.
3. If you get a Device is Active warning, close the Display pane or click the Disconnect button and right click the camera again.

4. Browse to select the firmware file and click Open.
5. Click Yes to continue.

To upgrade the firmware using FlyCapture:

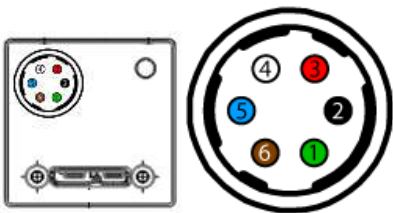
1. **Start Menu-->All Programs-->Point Grey FlyCapture2 SDK-->Utilities-->UpdaterGUI**
2. Select the camera from the list at the top.
3. Click Open to select the firmware file.
4. Click Update.
5. Click Yes to continue.

6 Input/Output Control

6.1 General Purpose Input/Output (GPIO)

The camera is equipped with a 6-pin GPIO connector on the back of the case. The connector is a Hirose HR10A-7R-6PB, the mating connector is a Hirose HR10A-7P-6S(73).

Table 6.1: GPIO pin assignments (as shown looking at rear of camera)

Diagram	Color	Pin	Function	Description
	Green	1	V _{EXT}	+12 V DC Camera Power
	Black	2	I0	Opto-isolated input (GPIO 0)
	Red	3	NC / +3.3 V	+3.3 V output. Current 120 mA (nominal). Firmware enabled (See OUTPUT_VOLTAGE_ENABLE: 19D0h)
	White	4	O1	Opto-isolated output (GPIO 1)
	Blue	5	OPTO_GND	Ground for opto-isolated I/O, not connected to camera ground
	Brown	6	GND	DC camera power ground

Power can be provided through the GPIO interface. The camera selects whichever power source is supplying a higher voltage.

Note: Some systems, especially those with laptop computers or longer cable lengths, may not provide adequate power through the USB 3.1 cable which could result in intermittent operation. The use of external power through the GPIO is required for these systems.

For more information on camera power, see [Powering the Camera](#).

For details on GPIO circuits, see [GPIO Electrical Characteristics](#).

6.2 GenICam Digital Input/Output Control

Name	Display Name	Description	Value
LineSelector	+ Line Selector	Selects the physical line (or GPIO pin) of the external device connector to configure.	Line 0 Line 1 Line 2 Line 3
LineMode	Line Mode	Controls whether the physical line is used to Input or Output a signal. Choices are dependent on which line is selected.	Input Trigger Strobe Output
LineSource	Line Source	Selects which input or output signal to output on the selected line. Line Mode must be Output.	Exposure Active External Trigger Active
LineInverter	Line Inverter	Controls the inversion of the signal of the selected input or output line	True False
StrobeEnabled	Strobe Enabled	Enables/disables strobe	True False
UserOutputValue	User Output Value	Sets the value of the user output selector	True = High False = Low
LineDebounceTime	Line Debounce Time	Sets the value of the selected line debouncer time in microseconds	
LineStatus	Line Status	Returns the current status of the selected input or output line	True = High False = Low
LineStatusAll	Line Status All	Returns the current status of all available line signals at time of polling in a single bitfield	
StrobeDuration	Strobe Duration	Outputs a specified number of pulses with programmable high and low duration	
ExposureActiveMode	Exposure Active Mode	If the camera supports rolling shutter with a global reset, the strobe can be set to active from the start of exposure of the first line to the end of exposure on the last line (Any pixel) or from the start of exposure on the last line to the end of exposure on the first line (All pixels).	00: Line 1 Exposed 01: Any pixel exposed (only for global reset mode) 10: All pixels exposed (same as Line 1 for global reset mode)
3.3VEnable	3.3 V Enable	Enable or disable 3.3 V output	

6.3 Strobe Signal Delay

In free running mode, there is an inherent minimum pulse width which differs for active-high and active-low strobe. For active-high (turning the diode off), if the pulse is less than 25 μs it is filtered. For active-low (turning the diode on) the pulse is shorter, approximately 10 μs .

There is a fixed pulse width difference between the expected strobe width and the actual width. For active-high, the pulse width is approximately 25 μs shorter than expected. For active-low, the pulse width is approximately 25 μs longer than expected.

6.4 Programmable Strobe Output

The camera is capable of outputting a strobe pulse off select GPIO pins that are configured as outputs. The start of the strobe can be offset from either the start of exposure (free-running mode) or time of incoming trigger (external trigger mode). By default, a pin that is configured as a strobe output will output a pulse each time the camera begins integration of an image.

The duration of the strobe can also be controlled. Setting a strobe duration value of zero produces a strobe pulse with duration equal to the exposure (shutter) time.

Multiple GPIO pins, configured as outputs, can strobe simultaneously.

Connecting two strobe pins directly together is not supported. Instead, place a diode on each strobe pin.

The camera can also be configured to output a variable strobe pulse pattern. The strobe pattern functionality allows users to define the frames for which the camera will output a strobe. For example, this is useful in situations where a strobe should only fire:

- Every Nth frame (e.g. odd frames from one camera and even frames from another); or
- N frames in a row out of T (e.g. the last 3 frames in a set of 6); or
- Specific frames within a defined period (e.g. frames 1, 5 and 7 in a set of 8)

Related Knowledge Base Articles

Title	Article
Buffering a GPIO pin strobe output signal using an optocoupler to drive external devices	Knowledge Base Article 10585
GPIO strobe signal continues after isochronous image transfer stops	Knowledge Base Article 10078

6.5 Debouncer

By default, our cameras reject a trigger signal that has a pulse width of less than 16 ticks of the pixel clock. With the debouncer you can define a debounce value. Once the debouncer is enabled and defined, the camera rejects a trigger signal with a pulse width less than the defined debounce value.

We recommend you set the debounce value slightly higher than longest expected duration of an invalid signal to compensate for the quality of the input clock signal.

The debouncer is available on GPIO input pins. For the debouncer to take effect, the associated GPIO pin must be in Input mode (GPIO Mode 0). The debouncer works in all trigger modes.

Note: Each GPIO has its own input delay time. The debouncer time adds additional delay to the signal on the pin.

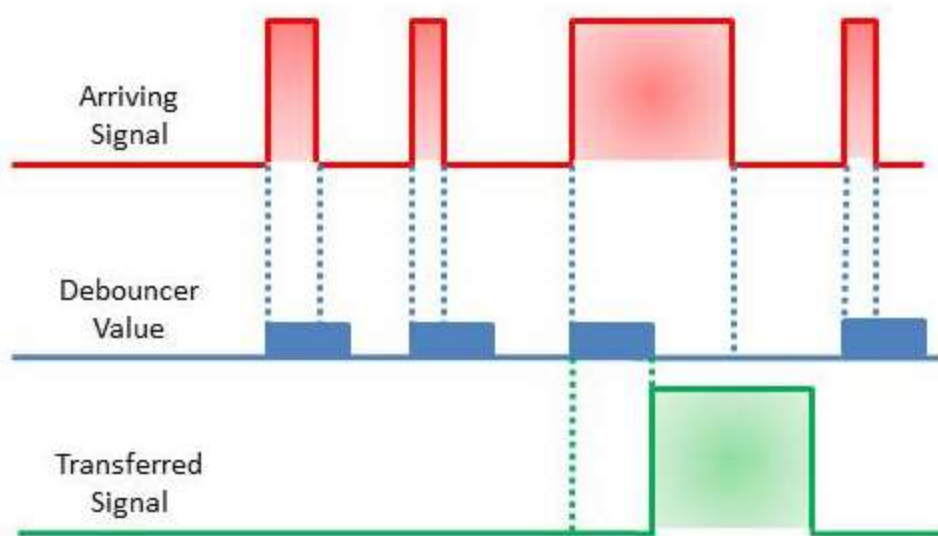


Figure 6.1: Debouncer Filtering Invalid Signals

To set the debouncer:

- GenICam—[Digital Input Output Control](#)

6.6 GPIO Electrical Characteristics

Both the opto-isolated input and output have over current protection.

The output is open collector and thus requires a pull-up resistor to operate. The rise time and bias current will be determined by the resistor value chosen. If the camera is generating an output signal that approaches the rise time plus the fall time of the opto-isolated circuit, care must be taken to optimize the pull-up resistor chosen to minimize the rise time while still remaining within the current limits of the output circuit.

The opto-isolated specifications listed below are applicable when power to the camera is provided through the interface and not through the GPIO.

Warning! To avoid damage, connect the OPTO_GND pin first before applying voltage to the GPIO line.

Warning! Prolonged use of the camera outside of the Operating Range described below may lead to unexpected behavior and should be avoided.

Table 6.2: Operating Range

Description	Minimum	Maximum
Opto-isolated Input Voltage	0 V	30 V
Opto-isolated Output Voltage	0 V	48 V
Opto-isolated Output Current		25 mA
3.3 V Output Current		120 mA

Table 6.3: Absolute Maximum Ratings

Description	Minimum	Maximum
Opto-isolated Input Voltage	-70 V	40 V
Opto-isolated Output Voltage	-48 V	48 V

Table 6.4: External Voltage Resistor Combinations

External Voltage	External Resistor	OPTO_OUT Low Voltage	OPTO_OUT High Voltage	Current
3.3 V	1 k Ω	0.6 V	3.3 V	2.8 mA
5 V	1 k Ω	0.8 V	5.0 V	4.2 mA
12 V	2.4 k Ω	0.9 V	12.0 V	4.6 mA
24 V	4.7 k Ω	0.9 V	24.0 V	4.9 mA

External Voltage	External Resistor	OPTO_OUT Low Voltage	OPTO_OUT High Voltage	Current
30 V	4.7 kΩ	1.1 V	30.0 V	6.2 mA
Values are for reference only				

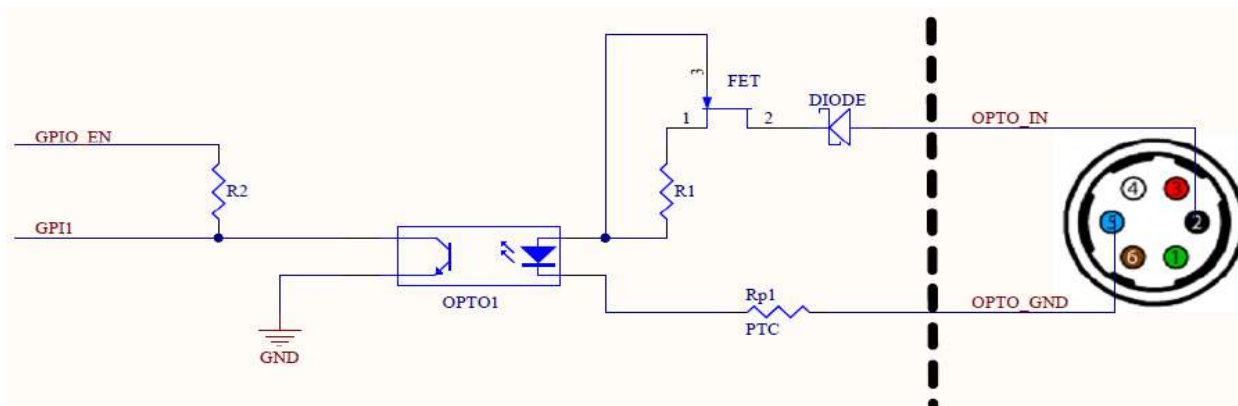


Figure 6.2: Opto-isolated input circuit

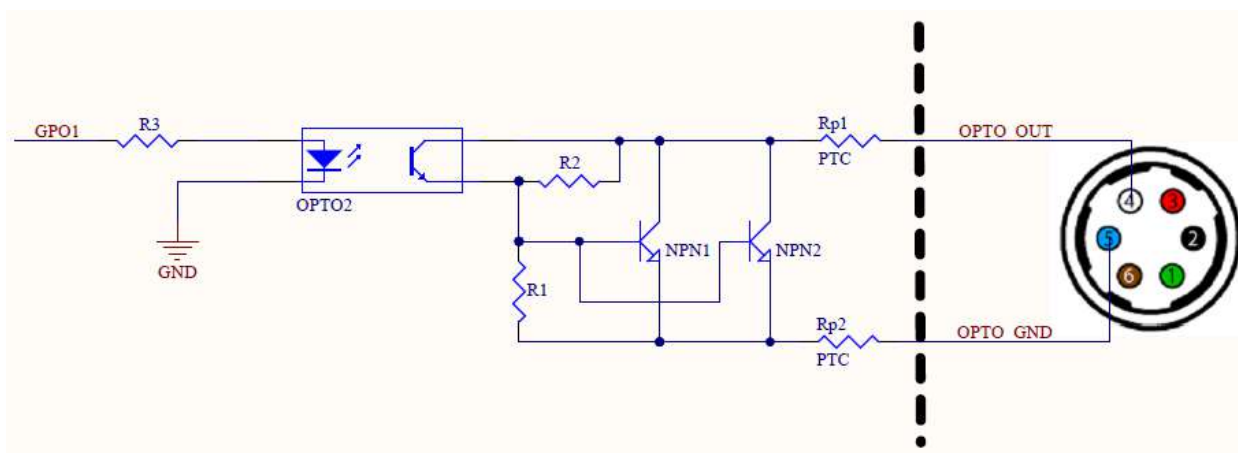


Figure 6.3: Opto-isolated output circuit

6.6.1 Output Timing Characteristics

Note: Timing specifications are preliminary and subject to change.

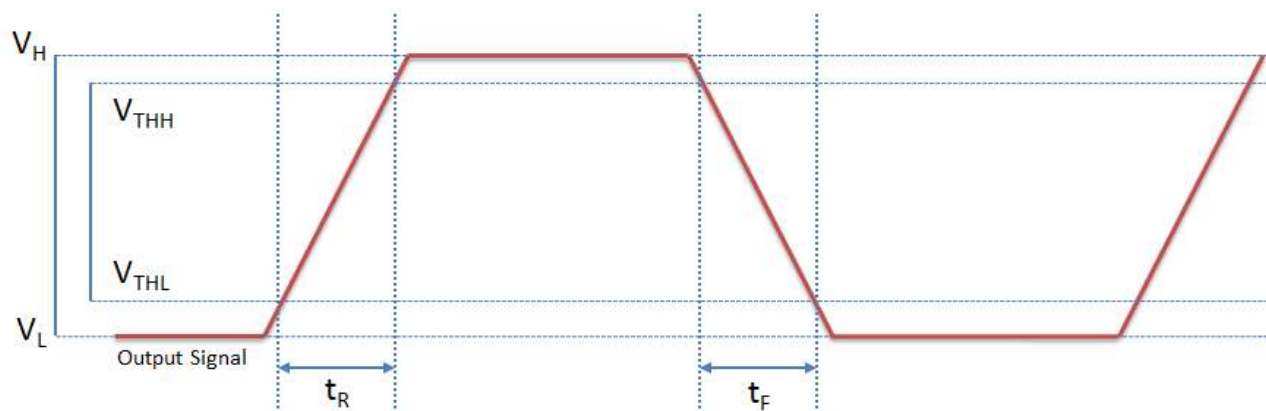


Figure 6.4: Output Timing Characteristics

Table 6.5: Opto-isolated Output Performance (measured at $V_{CC} = 3.3\text{ V}$, $R_{ext} = 200\ \Omega$)

Parameter	Symbol	Opto-isolated	
		Load	No Load
Output Low Voltage	V_L	1.3875 V	1.41 V
Output High Voltage	V_H	2.80625 V	3.26 V
Output Threshold High Voltage	V_{THH}	2.66 V	3.08 V
Output Threshold Low Voltage	V_{THL}	1.53 V	1.60 V
Cycle Rise Time	t_R	5.04 μs	6 μs
Cycle Fall Time	t_F	3.18 μs	3.66 μs
Opto Current		2.094 mA	7.96 mA
Opto Isolator Delay (High to Low)		5.34 μs	5.5 μs
Opto Isolator Delay (Low to High)		12.7 μs	15.24 μs

Table 6.6: Opto-isolated Output Performance (measured at $V_{CC} = 5\text{ V}$, $R_{ext} = 1\text{ k}\Omega$)

Parameter	Symbol	Opto-isolated	
		Load	No Load
Output Low Voltage	V_L	0.85 V	0.86 V
Output High Voltage	V_H	2.9 V	5.06 V
Output Threshold High Voltage	V_{THH}	2.695 V	4.64 V
Output Threshold Low Voltage	V_{THL}	1.063 V	1.28 V
Cycle Rise Time	t_R	11.4 μs	8.92 μs
Cycle Fall Time	t_F	1.84 μs	1.98 μs
Opto Current		2 mA	4.06 mA

Parameter	Symbol	Opto-isolated	
		Load	No Load
Opto Isolator Delay (High to Low)		3.8776 μs	4.117 μs
Opto Isolator Delay (Low to High)		26.754 μs	25.124 μs

Table 6.7: Sample Opto-isolated Output Voltage/Current Measurements

External Voltage	External Resistor	V_L	I_{OUT}
3.3 V	1 k Ω	575 mV	2.7 mA
5 V	1 k Ω	840 mV	4.1 mA
12 V	2.4 k Ω	915 mV	4.6 mA
24 V	4.7 k Ω	975 mV	4.9 mA

6.6.2 Input Timing Characteristics

Note: Timing specifications are preliminary and subject to change.

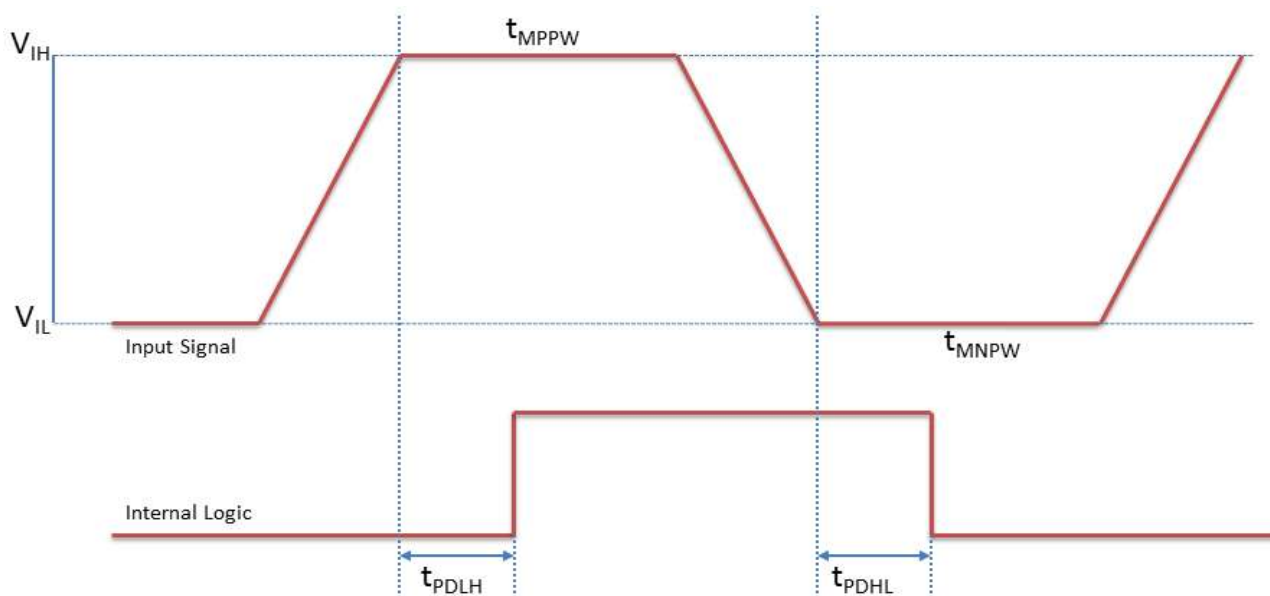


Table 6.8: Opto-isolated Input Performance (V_{CC} = varying pk-pk)

Parameter	Symbol	Opto-isolated with Load
Input Low Voltage	V_{IL}	$\leq 2 \text{ V}$
Input High Voltage	V_{IH}	$\geq 2.05 \text{ V}$
Propagation Delay Low to High	t_{PDLH}	19.74 μs

Parameter	Symbol	Opto-isolated with Load
Propagation Delay High to Low	t_{PDHL}	16.692 μ s
Typical Positive Pulse Width	t_{MPPW}	12 μ s
Typical Negative Pulse Width	t_{MNPW}	6 μ s

7 Image Acquisition

7.1 Asynchronous Triggering

The camera supports asynchronous triggering, which allows the start of exposure (shutter) to be initiated by an external electrical source (or hardware trigger) or (software trigger).

To access trigger modes:

- GenICam—[Acquisition Control](#)
- FlyCapture API—[AsyncTriggerEx](#)
- CSRs—[TRIGGER_MODE: 830h](#)

Model	Supported Acquisition Modes	Firmware (Mono / Color)
BFLY-U3-03S2	Standard, bulb, low smear, overlapped, multi-shot	1.8.3.0 / 1.8.3.0
BFLY-U3-05S2	Standard, bulb, low smear, overlapped, multi-shot	1.4.3.1 / 1.4.3.1
BFLY-U3-13S2	Standard, bulb, low smear, overlapped, multi-shot	1.1.3.0 / 1.1.3.0
BFLY-U3-20S4	Standard, bulb, low smear, overlapped, multi-shot	1.6.3.0 / 1.6.3.0
BFLY-U3-23S6	Standard, bulb, overlapped, multi-shot	1.8.3.0 / 1.8.3.0
BFLY-U3-50H5	Standard, bulb, low smear, overlapped, multi-shot	1.8.3.0 / 1.8.3.0
For More Information...		
Standard External Trigger (Mode 0)		
Bulb Shutter Trigger (Mode 1)		
Low Smear Trigger (Mode 13)		
Overlapped Exposure Readout Trigger (Mode 14)		
Multi-Shot Trigger (Mode 15)		

7.1.1 GenICam Acquisition Control

Name	Display Name	Description	Value
AcquisitionMode	Acquisition Mode	Sets the acquisition mode of the device	Continuous Single Frame Multi Frame
AcquisitionStart	Acquisition Start	Starts the acquisition of the device	Write Only
AcquisitionStop	Acquisition Stop	Stops the acquisition of the device at the end of the current frame	Write Only
AcquisitionFrameCount	Acquisition Frame Count	Number of frames to acquire in Multi Frame acquisition mode	

Name	Display Name	Description	Value
AcquisitionFrameRate	Acquisition Frame Rate (Hz)	Controls the acquisition rate (in Hertz) at which the frames are captured	
AcquisitionFrameRateControlEnabled	Acquisition Frame Rate Control Enabled	Enables manual control of the camera frame rate	True False
FrameRateAuto	Frame Rate Auto	Controls the mode for automatic frame rate adjustment	Off Continuous
TriggerSelector	Trigger Selector	Selects the type of trigger to configure. Derived from Exposure Mode.	Exposure Start/ Exposure Active
TriggerMode	Trigger Mode	Controls whether or not the selected trigger is active	Off On
TriggerSource	Trigger Source	Specifies the internal signal or physical input line to use as the trigger source. The selected trigger must have its Trigger Mode set to On.	Software Line x where x is a GPIO trigger pin
TriggerActivation	Trigger Activation	Specifies the activation mode of the trigger	Falling Edge Rising Edge
TriggerDelay	TriggerDelay (us)	Specifies the delay (in microseconds) to apply after the trigger reception before activating it	
TriggerDelayEnabled	Trigger Delay Enabled	Specifies whether or not the Trigger Delay is enabled	True False
ExposureMode	Exposure Mode (not all models support all modes)	Sets the operation mode of the exposure (shutter). Toggles the Trigger Selector. Timed = Exposure Start; Trigger Width = Exposure Active	Timed Trigger Width
ExposureTime	ExposureTime (us)	Exposure time in microseconds when Exposure Mode is Timed	
ExposureAuto	Exposure Auto	Sets the automatic exposure mode when Exposure mode is Timed	Off Once Continuous
pgrExposureCompensationAuto	Exposure Compensation Auto	Sets the automatic exposure compensation value mode	
pgrExposureCompensation	Exposure Compensation	The measured or target image plane illuminance in EV	
pgrAutoExposureCompensationLowerLimit	Auto Exposure Compensation Lower Limit	Lower limit of the auto exposure compensation value(EV) parameter	
pgrAutoExposureCompensationUpperLimit	Auto Exposure Compensation Upper Limit	Upper limit of the auto exposure compensation value(EV) parameter	

7.1.2 Standard External Trigger (Mode 0)

Trigger Mode 0 is best described as the standard external trigger mode. When the camera is put into Trigger Mode 0, the camera starts integration of the incoming light from external trigger input falling/rising edge. The Exposure Time describes integration time. No parameter is required. The camera can be triggered in this mode by using the GPIO pins as external trigger or by using a software trigger.

It is not possible to trigger the camera at full frame rate using Trigger Mode 0; however, [Overlapped Exposure Readout Trigger \(Mode 14\)](#) allows triggering at nearly full frame rate.

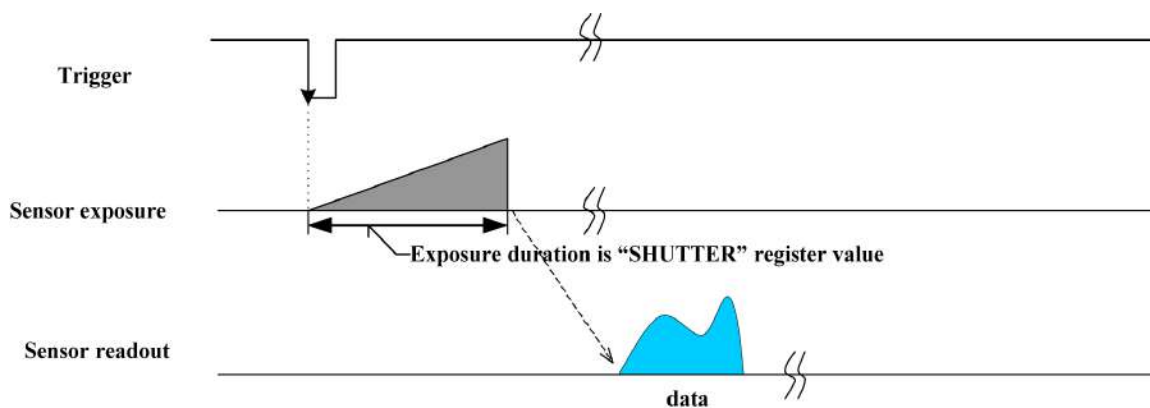


Figure 7.1: Trigger Mode 0 ("Standard External Trigger Mode")

GenICam—Acquisition Control	
Acquisition Mode	Continuous
Trigger Selector	Exposure Start
Trigger Mode	On
Trigger Source	Line x (GPIO pin)
Trigger Activation	Rising or Falling edge
Trigger Delay	0
Exposure Mode	Timed
Exposure Time	Integration Time
Exposure Auto	Off

Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_0
Parameter	[20-31]	None

7.1.3 Bulb Shutter Trigger (Mode 1)

In Bulb Shutter mode, the camera starts integration of the incoming light from external trigger input. Integration time is equal to low state time of the external trigger input.

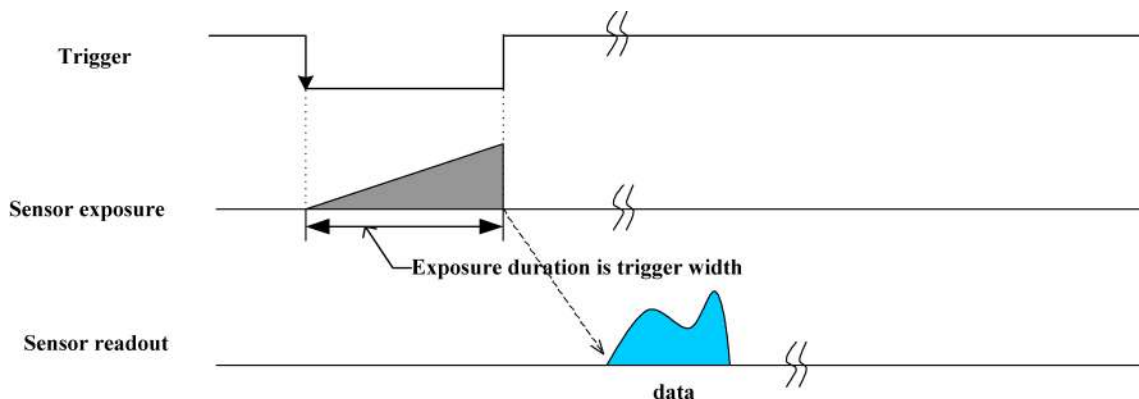


Figure 7.2: Trigger Mode 1 (“Bulb Shutter Mode”)

GenICam—Acquisition Control	
Acquisition Mode	Trigger width
Trigger Selector	Exposure Active
Trigger Mode	On
Trigger Source	Line x (GPIO pin)
Trigger Activation	Rising or Falling edge
Trigger Delay	0
Exposure Mode	Trigger Width
Exposure Time	Integration Time
Exposure Auto	Off

Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_1
Parameter	[20-31]	None

7.1.4 Low Smear Trigger (Mode 13)

Trigger Mode 13 is a reduced smear imaging mode.

Smear reduction works by increasing the speed of the vertical clock near the end of the integration cycle. This step is also known as fast dump. Since the clock speed has been increased, this reduces the time each pixel data has to collect smear while it passes through the vertical shift register.

This trigger mode behaves similarly to [Standard External Trigger \(Mode 0\)](#), except the trigger input first activates a fast dump off the CCD. The fast dump period is followed by exposure, which is controlled by the Shutter settings. The length of the fast dump period is determined by the trigger delay.

For other methods to minimize smear, see [Smear Reduction](#).

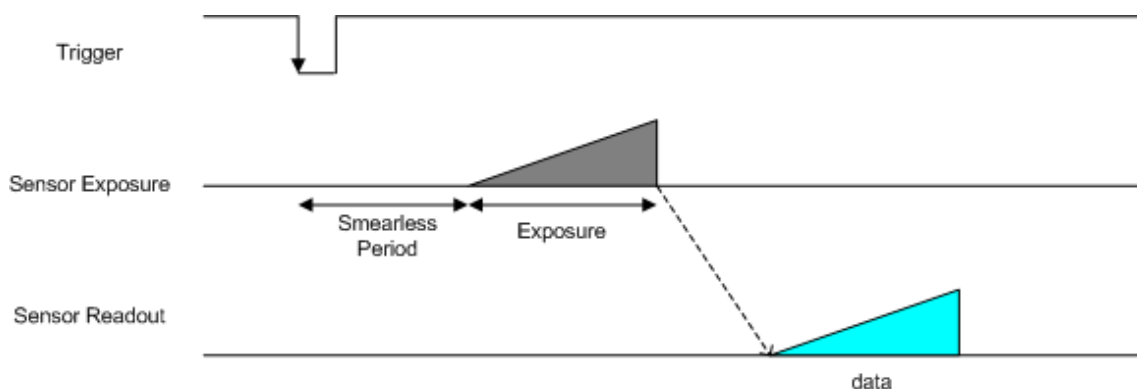


Figure 7.3: Trigger Mode 13 (“Low Smear Trigger Mode”)

Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_13
Parameter	[20-31]	None

7.1.5 Overlapped Exposure Readout Trigger (Mode 14)

Overlapped Exposure Readout mode is a vendor-unique trigger mode that is very similar to Standard External mode, but allows for triggering at faster frame rates. This mode works well for users who want to drive exposure start with an external event. However, users who need a precise exposure start should use [Standard External Trigger \(Mode 0\)](#).

In the figure below, the trigger may be overlapped with the readout of the image, similar to continuous shot (free-running) mode. If the trigger arrives after readout is complete, it starts as quickly as the imaging area can be cleared. If the trigger arrives before the end of shutter integration (that is, before the trigger is *armed*), it is dropped. If the trigger arrives while the image is still being read out of the sensor, the start of exposure is delayed until the next opportunity to clear the imaging area without injecting noise into the output image. The end of exposure cannot occur before the end of the previous image readout. Therefore, exposure start may be delayed to ensure this, which means priority is given to maintaining the proper exposure time instead of to the trigger start.

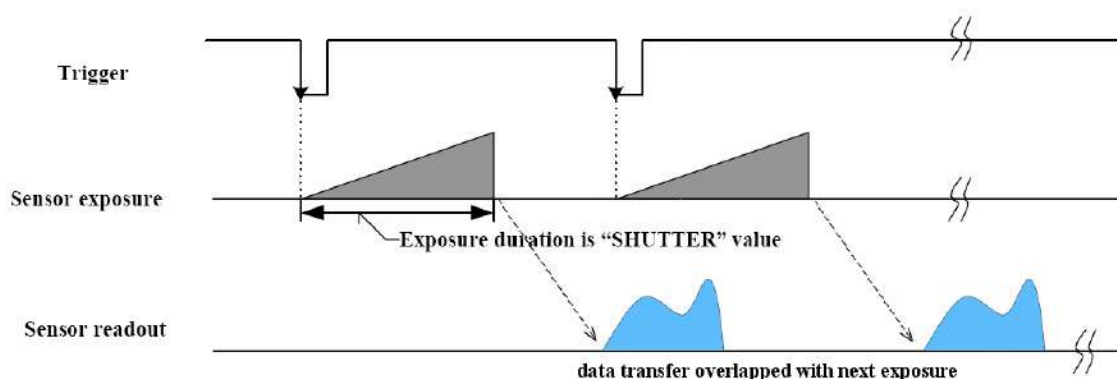


Figure 7.4: Trigger Mode 14 (“Overlapped Exposure/Readout Mode”)

Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_14
Parameter	[20-31]	None

7.1.6 Multi-Shot Trigger (Mode 15)

Multi-Shot mode is a vendor-unique trigger mode that allows the user to fire a single hardware or software trigger and have the camera acquire and stream a predetermined number of images.

The number of images to be acquired is determined by the parameter specified with the trigger mode. This allows up to 255 images to be acquired from a single trigger. Setting the parameter to 0 results in a non-free running, non-overlap mode. If Trigger Mode 14 is supported by the camera, setting the parameter to 0 results in a non-free running, overlap mode.

Once the trigger is fired, the camera will acquire N images with an exposure time equal to the value defined by the shutter, and stream the images to the host system at the current frame rate. Once this is complete, the camera can be triggered again to repeat the sequence.

Any changes to the trigger control cause the current sequence to stop.

Note: During the capture of N images, provided that $N > 0$, the camera is still in an asynchronous trigger mode (essentially Trigger Mode 14), rather than continuous (free-running) mode. The result of this is that the frame rate is turned OFF, and the camera put into extended shutter mode. Users should ensure that the maximum shutter time is limited to $1/\text{frame_rate}$ to get the N images captured at the current frame rate.

Related Knowledge Base Articles

Title	Article
Extended shutter mode operation for FLIR machine vision cameras	Knowledge Base Article 10087

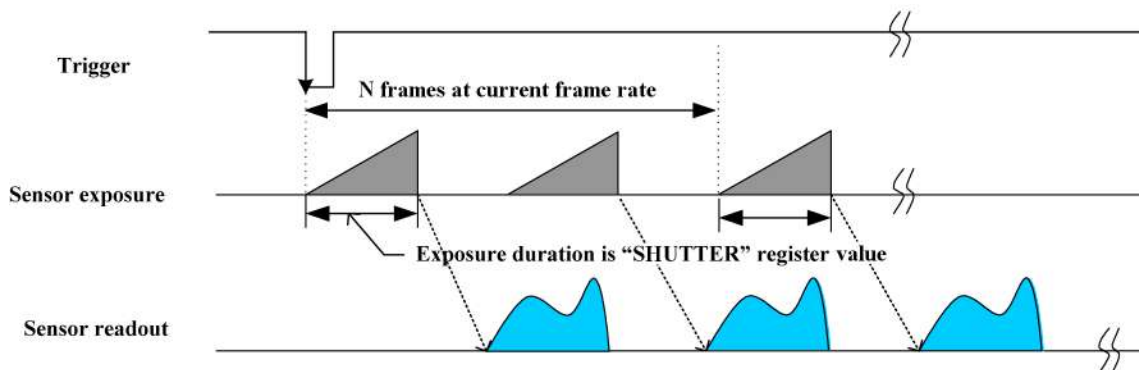


Figure 7.5: Trigger Mode 15, when N is greater than 0. ("Multi-Shot Trigger Mode")

Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High

Registers—TRIGGER_MODE: 830h		
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_15
Parameter	[20-31]	<i>N</i> number of images to be acquired

7.2 External Trigger Timing

The time from the external trigger firing to the start of shutter is shown below:

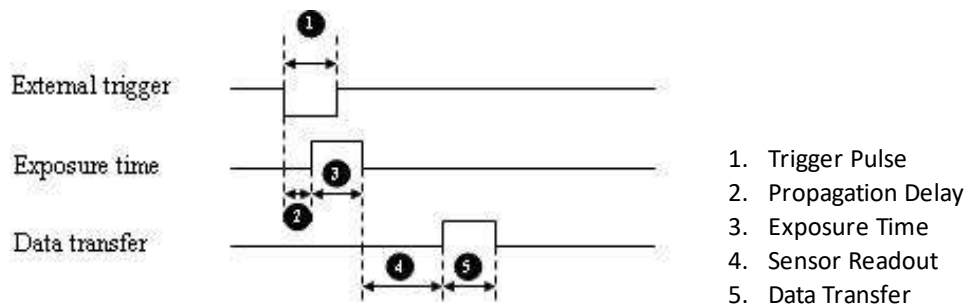


Figure 7.6: External trigger timing characteristics

Depending on the firmware version, Sensor Readout (4) and Data Transfer (5) may occur concurrently.

It is possible for users to measure this themselves by configuring one of the camera's GPIO pins to output a strobe pulse (see [Programmable Strobe Output](#)) and connecting an oscilloscope up to the input trigger pin and the output strobe pin. The camera will strobe each time an image acquisition is triggered; the start of the strobe pulse represents the start of exposure.

7.3 Camera Behavior Between Triggers

When operating in external trigger mode, the camera clears charges from the sensor at the horizontal pixel clock rate determined by the current frame rate. For example, if the camera is set to 10 FPS, charges are cleared off the sensor at a horizontal pixel clock rate of 15 KHz. This action takes place following shutter integration, until the next trigger is received. At that point, the horizontal clearing operation is aborted, and a final clearing of the entire sensor is performed prior to shutter integration and transmission.

7.4 Changing Video Modes While Triggering

You can change the video format and mode of the camera while operating in trigger mode. Whether the new mode that is requested takes effect in the next triggered image depends on the timing of the request and the trigger mode in effect. The diagram below illustrates the relationship between triggering and changing video modes.

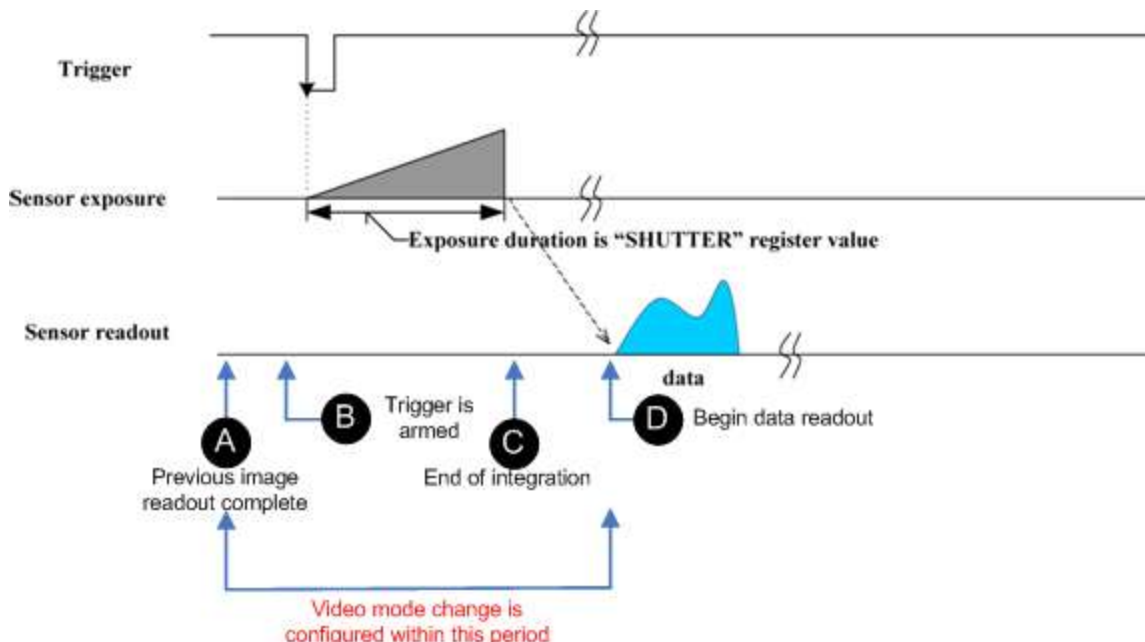


Figure 7.7: Relationship Between External Triggering and Video Mode Change Request

When operating in [Standard External Trigger \(Mode 0\)](#) or in [Bulb Shutter Trigger \(Mode 1\)](#), video mode change requests made before point A on the diagram are honored in the next triggered image. The camera attempts to honor a request made after point A in the next triggered image, but this attempt may or may not succeed, in which case the request is honored one triggered image later. In [Overlapped Exposure Readout Trigger \(Mode 14\)](#), point B occurs before point A. The result is that, in most cases, there is a delay of one triggered image for a video mode request, made before the configuration period, to take effect. In [Multi-Shot Trigger \(Mode 15\)](#), change requests made after point A for any given image readout are honored only after a delay of one image.

7.5 Asynchronous Software Triggering

Shutter integration can be initiated by a software trigger by setting the Trigger Source to Software in the GenICam features.

The time from a software trigger initiation to the start of shutter is shown below:

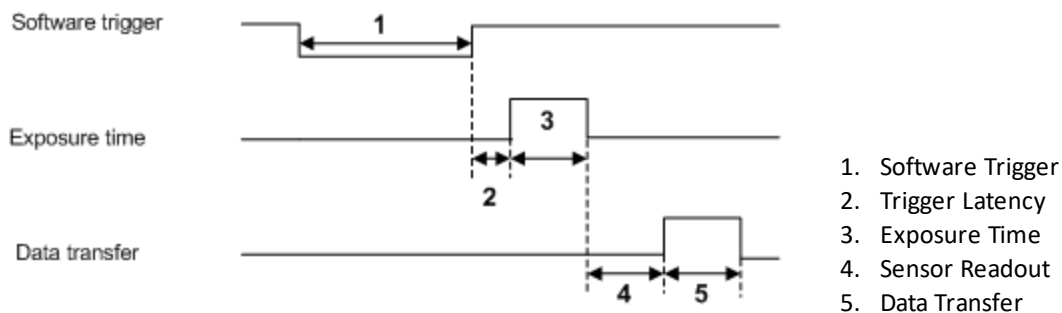


Figure 7.8: Software trigger timing

The time from when the software trigger is written on the camera to when the start of integration occurs can only be approximated. We then add the trigger latency (time from the trigger pulse to the start of integration) to this.

Note: This timing is solely from the camera perspective. It is virtually impossible to predict timing from the user perspective due to latencies in the processing of commands on the host PC.

8 Blackfly USB 3.1 Attributes

8.1 Pixel Formats

Pixel formats are an encoding scheme by which color or monochrome images are produced from raw image data. Most pixel formats are numbered 8, 12, or 16 to represent the number of bits per pixel.

The Blackfly USB 3.1's [Analog-to-Digital Converter](#), which digitizes the images, is configured to a fixed bit output (12-bit; 10- and 12-bit (BFLY-U3-23S6)). If the pixel format selected has fewer bits per pixel than the ADC output, the least significant bits are dropped. If the pixel format selected has greater bits per pixel than the ADC output, the least significant bits are padded and can be discarded by the user.

Pixel Format	Bits per Pixel
Mono 8, Raw 8	8
Mono 12, Raw 12, YUV 411	12
Mono 16, Raw 16, YUV 422	16
RGB 8, YUV 444	24

8.1.1 Raw

Raw is a pixel format where image data is Bayer RAW untouched by any on board processing. Selecting a Raw format bypasses the FPGA/color core, which may disable image processing, but allows for faster frame rates. In a GenICam application, this is achieved using Image Format Control and disabling On Board Color Processing.

8.1.2 Mono

Mono is a pixel format where image data is monochrome. Color cameras using a mono format enable FPGA/color core image processing such as access to gamma/LUT.

Y8 and Y16 are also monochrome formats with 8 and 16 bits per pixel respectively.

8.1.3 RGB

RGB is a color-encoding scheme that represents the intensities of red, green, and blue channels in each pixel. Each color channel uses 8 bits of data. With 3 color channels, a single RGB pixel is 24 bits.

8.1.4 YUV

YUV is a color-encoding scheme that assigns both brightness (Y) and color (UV) values to each pixel. Each Y, U, and V value comprises 8 bits of data. Data transmission can be in 24, 16, or 12 bits per pixel. For 16 and 12 bits per pixel transmissions, the U and V values are shared between pixels to free bandwidth and possibly increase frame rate.

YUV444 is considered a high resolution format which transmits 24 bits per pixel. Each Y, U, and V value has 8 bits.

YUV422 is considered a medium resolution format which transmits 16 bits per pixel. Each Y value has 8 bits, but the U and V values are shared between 2 pixels. This reduces the bandwidth of an uncompressed video signal by one-third with little to no visual difference.

YUV411 is considered a low resolution format which transmits 12 bits per pixel. Each Y value has 8 bits, but the U and V values are shared between 4 pixels. This reduces bandwidth by one half compared to YUV444, but also reduces the color information being recorded.

YUV can be either packed or planar. Packed is when the Y, U, and V components are stored in a single array (macropixel). Planar is when the Y, U, and V components are stored separately and then combined to form the image. FLIR machine vision cameras use packed YUV.

Related Knowledge Base Articles

Title	Article
Understanding YUV data formats	Knowledge Base Article 10092

8.2 Video Modes Overview

On FLIR machine vision cameras, binning refers to the aggregation of pixels. Analog binning is aggregation that occurs before the analog to digital conversion. Digital binning is aggregation that occurs after the analog to digital conversion. Unless specified otherwise, color data is maintained in binning modes.

In most cases, pixels are added once they are binned. Additive binning usually results in increased image intensity. Another method is to average the pixel values after aggregation. Binning plus averaging results in little or no change in the overall image intensity.

Subsampling, or decimation, refers to the skipping of pixels.

Binning and subsampling reduces the effective image resolution. For example, 2x2 binning reduces both the width and the height by a quarter.

The figures below illustrate binning and subsampling. 2x vertical binning aggregates two adjacent vertical pixel values to form a single pixel value. 2x horizontal binning works in the same manner, except two adjacent horizontal pixel values are aggregated. 2x2 subsampling skips every second pixel horizontally and vertically.

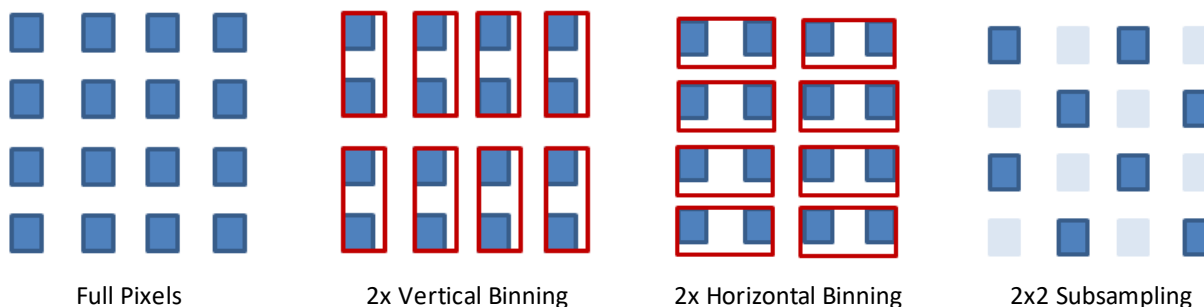


Figure 8.1: Aggregation and Decimation of Pixels

Moving the ROI position to a different location does not require the camera to be stopped and restarted, unless the change is illegal (e.g. moving the ROI outside the imaging area).

Note: [Pixel correction](#) is not done in any of the binning modes

8.2.1 Blackfly USB 3.1 Video Mode Descriptions

Mode	Models	Description	Frame Rate Increase	Brightness Increase
0	All	All Pixel Scan	N/A	N/A
1	BFLY-U3-03S2M-CS BFLY-U3-05S2M-CS BFLY-U3-20S4M-CS BFLY-U3-50H5M-C BFLY-U3-13S2M-CS	2x2 Adjacent Binning	Yes	Yes
	BFLY-U3-03S2C-CS BFLY-U3-05S2C-CS BFLY-U3-13S2C-CS BFLY-U3-20S4C-CS BFLY-U3-23S6M-C BFLY-U3-50H5C-C	2x2 Bayer Binning	No	Yes
4	BFLY-U3-03S2C-CS BFLY-U3-05S2C-CS BFLY-U3-13S2C-CS BFLY-U3-20S4C-CS BFLY-U3-50H5C-C	2x2 Adjacent Binning	Yes	No
	BFLY-U3-23S6C-C	2x2 Bayer Binning	No	Yes
5	BFLY-U3-03S2M-CS BFLY-U3-05S2M-CS BFLY-U3-13S2M-CS BFLY-U3-20S4M-CS BFLY-U3-50H5M-C	4x4 Adjacent Binning	Yes	Yes
	BFLY-U3-20S4C-CS	4x4 Adjacent Binning	No	Yes
	BFLY-U3-03S2C-CS BFLY-U3-05S2C-CS BFLY-U3-13S2C-CS BFLY-U3-50H5C-C	4x4 Bayer Binning	No	Yes
7	BFLY-U3-23S6M-C BFLY-U3-23S6C-C	Optimal Imaging	No	No

Mode 0

Mode 0 is the standard imaging mode with full resolution readout, and global shutter.

Mode 1

This is 2x2 binning. Implementation and impact on frame rate varies between models. Effective resolution is reduced by half and image brightness is increased in all cases.

Monochrome CCD models implement this binning mode vertically on the sensor and horizontally in the FPGA. There is an increase in image brightness and frame rate.

Color CCD models implement this binning mode in the FPGA entirely. There is an increase in brightness but no impact on frame rate.

CMOS models implement binning mode in the sensor as well. There is an increase in brightness and frame rate in some models.

Mode 4

Mode 4 is 2x2 binning.

On color CCD cameras, vertical binning is done on the sensor and horizontal binning in the FPGA. There is an increase in image brightness and frame rate. Compared to Mode 1, this mode suffers from reduced image quality resulting from vertical sensor binning and inherent skipping of rows during readout. Applications using color CCD cameras should use this mode if a higher frame is required.

Color CMOS cameras perform vertical and horizontal binning in the FPGA. There is an increase in brightness, but not in frame rate.

Mode 5

Mode 5 is 4x4 binning. Implementation and impact on frame rate varies between models. Effective resolution is reduced by a factor of four and image brightness is increased in all cases.

Monochrome CCD models implement this binning mode vertically on the sensor and horizontally in the FPGA. There is an increase in image brightness and frame rate.

Color CCD models implement this binning mode in the FPGA entirely. There is an increase in brightness but no impact on frame rate.

The Raw pixel format is not available in mode 5.

Mode 7

For BFLY-U3-23S6—mode 7 is optimized for high well depth, SNR, and quantum efficiency at the expense of frame rate. This mode uses a 12-bit ADC.

8.3 GenICam Image Format Control

Name	Display Name	Description	Value
SensorWidth	Sensor Width	Effective width of the sensor in pixels	
SensorHeight	Sensor Height	Effective height of the sensor in pixels	
MaxWidth	Max Width	Maximum width of the image in pixels	
MaxHeight	Max Height	Maximum height of the image in pixels	
Width	Width	Width of the image provided by the device in pixels	
Height	Height	Height of the image provided by the device in pixels	
OffsetX	Offset X	Vertical offset from the origin to the AOI in pixels	
OffsetY	Offset Y	Horizontal offset from the origin to the AOI in pixels	
ReverseX	Reverse X	Flip horizontally the image sent by the device. The AOI is applied after the flip	True False
	On Board Color Process Enabled	Enable FPGA processing (i.e., Gamma/Lookup table). This limits frame rate.	True False
PixelFormat	Pixel Format	Format of the pixel data (not all cameras support all formats)	Mono8, Mono12, Mono16, Raw8, Raw12, Raw16, RGB, YUV411, YUV422
PixelCoding	Pixel Coding	Coding of the pixels in the image	Mono Raw YUV RGB
PixelSize	Pixel Size	Size of a pixel in bits	8/12/16/24
PixelColorFilter	Pixel Color Filter	Type of color filter that is applied to the image	
TestImageSelector	Test Image Selector	Selects the type of test image that is sent by the camera	Off Test Image 1 Test Image 2
VideoMode	Video Mode	Current video mode	0 ... 8
PixelBigEndian	Pixel BigEndian	Set the pixel endianness for pixel format Mono16	True False
BinningHorizontal	Binning Horizontal	Number of horizontal pixels to combine together	
BinningVertical	Binning Vertical	Number of vertical pixels to combine together	
PixelDynamicRangeMin	Dynamic Range Min	Indicates the minimum pixel value transferred from the camera	
PixelDynamicRangeMax	Dynamic Range Max	Indicates the maximum pixel value transferred from the camera	

Name	Display Name	Description	Value
OnBoardColorProcessing	On Board Color Processing	Disabling On Board Color Processing bypasses the FPGA/color core, which disables image processing, but allows for faster frame rates.	True False

8.4 Frame Rates

8.4.1 BFLY-U3-03S2 Frame Rates by Video Mode

Note: Frame rates are not affected by pixel format selection.

8.4.1.1 BFLY-U3-03S2M

Firmware: 1.3.3.0

Mode 0

648 x 488	320 x 240	160 x 120
84	129	177

Mode 1

324 x 244	160 x 120
146	190

Mode 5

160 x 122
228

8.4.1.2 BFLY-U3-03S2C

Firmware: 1.3.3.0

Mode 0

640 x 480	320 x 240	160 x 120
84	129	177

Mode 1

324 x 244	160 x 120
84	129

Mode 4

324 x 244	160 x 120
146	190

Mode 5

160 x 122
84

8.4.2 BFLY-U3-05S2 Frame Rates by Video Mode

Note: Frame rates are not affected by pixel format selection.

8.4.2.1 BFLY-U3-05S2M

Firmware: 1.6.3.0

Mode 0

808 x 608	320 x 240	160 x 120
50	88	117

Mode 1

404 x 304	160 x 120
83	116

Mode 5

200 x 122
120

8.4.2.2 BFLY-U3-05S2C

Firmware: 1.6.3.0

Mode 0

808 x 608	320 x 240	160 x 120
50	88	117

Mode 1

404 x 304	160 x 120
50	88

Mode 4

404 x 304	160 x 120
83	116

Mode 5

200 x 122
50

8.4.3 BFLY-U3-13S2 Frame Rates by Video Mode

Note: Frame rates are not affected by pixel format selection.

8.4.3.1 BFLY-U3-13S2M

Firmware: 1.2.3.2

Mode 0

1288 x 964	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
30	35	42	50	80	112

Mode 1

644 x 482	320 x 240	160 x 120
50	79	110

Mode 5

320 x 240	160 x 120
88	115

8.4.3.2 BFLY-U3-13S2C

Firmware: 1.2.3.2

Mode 0

1288 x 964	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
30	35	42	50	80	112

Mode 1

644 x 482	320 x 240	160 x 120
30	50	80

Mode 4

644 x 482	320 x 240	160 x 120
50	79	110

Mode 5

320 x 240	160 x 120
30	50

8.4.4 BFLY-U3-20S4 Frame Rates by Video Mode

Note: Frame rates are not affected by pixel format selection.

8.4.4.1 BFLY-U3-20S4M

Firmware: 1.6.3.0

Mode 0

1624 x 1224	1600 x 1200	1280 x 960	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
15	15	18	21	26	30	46	62

Mode 1

812 x 612	640 x 480	320 x 240	160 x 120
27	32	48	65

Mode 5

404 x 306	320 x 240	160 x 120
48	52	66

8.4.4.2 BFLY-U3-20S4C

Firmware: 1.6.3.0

Mode 0

1624 x 1224	1600 x 1200	1280 x 960	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
15	15	18	21	26	30	46	62

Mode 1

812 x 612	640 x 480	320 x 240	160 x 120
15	18	30	46

Mode 4

812 x 612	640 x 480	320 x 240	160 x 120
27	32	48	65

Mode 5

404 x 306	320 x 240	160 x 120
15	18	30

8.4.5 BFLY-U3-23S6 Frame Rates by Video Mode

Note: Except where noted, frame rates are not affected by pixel format selection.

8.4.5.1 BFLY-U3-23S6M

Firmware: 1.8.3.0

Mode 0

1920 x 1200	1280 x 960	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
41	51	63	79	97	177	299

Mode 1

960 x 600	640 x 480	320 x 240	160 x 120
41	51	97	177

Mode 7

1920 x 1200	1280 x 960	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
32	39	49	61	75	137	232

8.4.5.2 BFLY-U3-23S6C

Firmware: 1.8.3.0

Mode 0

1920 x 1200	1280 x 960	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
41 (28 in 24-bit)	51	63	79	97	177	299

Mode 4

960 x 600	640 x 480	320 x 240	160 x 120
41	51	97	177

Mode 7

1920 x 1200	1280 x 960	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
32 (28 in 24-bit)	39	49	61	75	137	232

8.4.6 BFLY-U3-50H5 Frame Rates by Video Mode

Note: Frame rates are not affected by pixel format selection.

8.4.6.1 BFLY-U3-50H5M

Firmware: 1.8.3.0

Mode 0

2448 x 2048	1920 x 1080	1280 x 960	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
7.5	11.5	12	13.9	15.6	17	21.5	24.6

Mode 1

1224 x 1024	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
12.49	14	15.6	16.8	19.9	22

Mode 5

612 x 512	320 x 240	160 x 120
20.11	20.11	20.11

8.4.6.2 BFLY-U3-50H5C

Firmware: 1.8.3.0

Mode 0

2448 x 2048	1920 x 1080	1280 x 960	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
7.5	11.5	12	13.9	15.6	17	21.5	24.6

Mode 1

1224 x 1024	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
7.5	9	10.5	12.3	17	21.5

Mode 4

1224 x 1024	1024 x 768	800 x 600	640 x 480	320 x 240	160 x 120
12.49	14	15.6	16.8	19.9	22

Mode 5

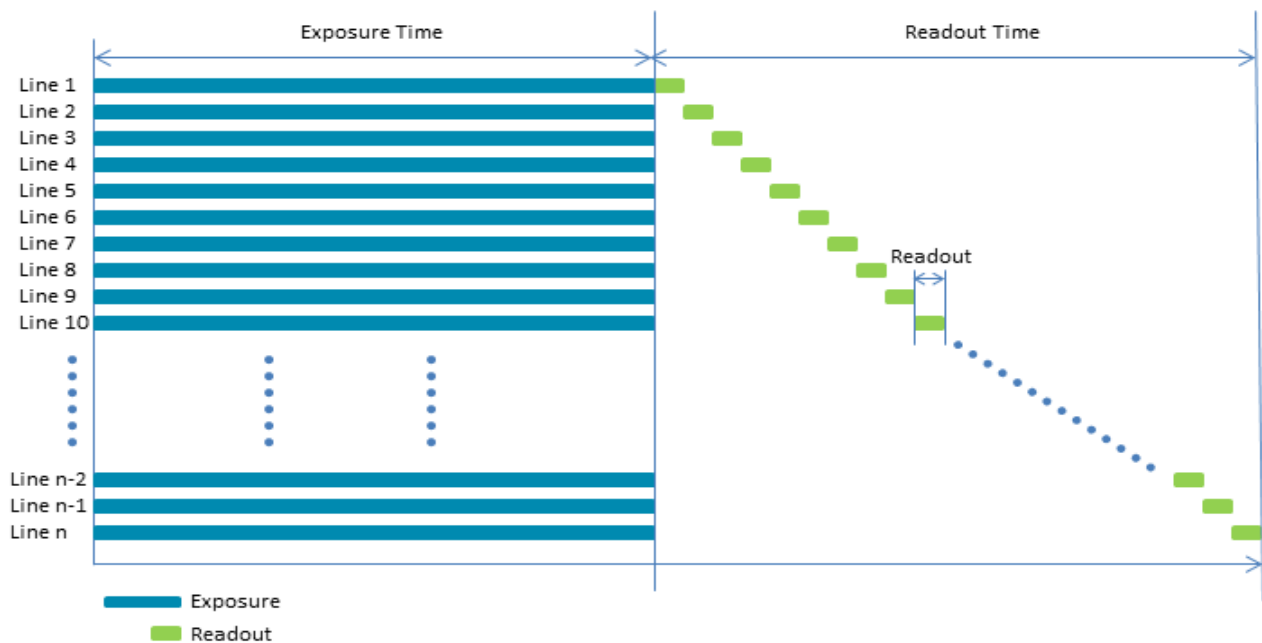
612 x 512	320 x 240	160 x 120
7.5	12.3	17

8.5 Readout Method (Shutter Type)

Model	Readout Method
BFLY-U3-03S2	Global shutter
BFLY-U3-05S2	Global shutter
BFLY-U3-13S2	Global shutter
BFLY-U3-20S4	Global shutter
BFLY-U3-23S6	Global shutter
BFLY-U3-50H5	Global shutter
For more information...	
Global Shutter	

8.5.1 Global Shutter

For cameras with a global shutter sensor, for each frame all of the lines start and stop exposure at the same time. The exposure time for each line is the same. Following exposure, data readout begins. The readout time for each line is the same but the start and end times are staggered. Readout time for a line is equal to $1/\text{Horizontal Line Frequency}$.



Some advantages of global shutter are more uniform brightness and minimal motion blur.

8.6 Overview of Imaging Parameters

The camera supports control over the following imaging parameters:

Imaging Parameter	GenICam Feature	FlyCapture API Sample Code
Brightness	Analog Control	Setting Brightness Using the FlyCapture API
Exposure Time	Acquisition Control	Setting Shutter Using the FlyCapture API
Gain	Analog Control	Setting Gain Using the FlyCapture API
Auto Exposure	Acquisition Control	Setting Auto Exposure Using the FlyCapture API
Sharpness	Analog Control	Setting Sharpness Using the FlyCapture API
Gamma and Lookup Table	Analog Control	Setting Gamma Using the FlyCapture API
Image Flip/Mirror	Image Format Control	
Embedded Image Information	Chunk Data Control	
Bayer Color Processing (color models only)	Image Format Control	Accessing Raw Bayer Data using FlyCapture
Hue (color models only)	Analog Control	Setting Hue Using the FlyCapture API
Saturation (color models only)	Analog Control	Setting Saturation Using the FlyCapture API

Most of these imaging parameters are defined by **modes** and **values**.

There are three modes:

GenICam Control

Mode	Description
Off	Feature is in manual mode and values can be set
Continuous	Feature is in automatic mode and values cannot be set
Once	Feature executes once automatically and then returns to manual mode

Note: The term Continuous is the same as Auto and the term Once is the same as One Push.

Users can define the values for manual operation of a feature.

8.7 GenICam Analog Control

Name	Display Name	Description	Value
Gain	Gain (dB)	Gain applied to the image in dB	
GainAuto	Gain Auto	Controls the mode for automatic gain adjustment	Off Once Continuous
BlackLevel	Black Level (percent)	Analog black level (brightness) in percent	
BlackLevelEnabled	Black Level Enabled	Enables/disables black level adjustment	True False
	Black Level Auto	Controls the mode for automatic black level adjustment	Off Once Continuous
BalanceRatioSelector	Balance Ratio Selector	Selects which balance ratio to control (for White Balance)	Red Blue
BalanceRatio	Balance Ratio	Controls the ratio of the selected color component to a reference color component	
BalanceWhiteAuto	Balance White Auto	Controls the mode for automatic white balancing between color channels	Off Once Continuous
Gamma	Gamma	Controls the gamma correction of pixel intensity	
GammaEnabled	Gamma Enabled	Enables/disables gamma correction	True False
Sharpness	Sharpness	Sharpness of the image	
SharpnessEnabled	Sharpness Enabled	Enables/disables sharpness adjustment	True False
SharpnessAuto	Sharpness Auto	Controls the mode for automatic sharpness adjustment	Off Once Continuous
Hue	Hue (degrees)	Hue of the image in degrees	
HueEnabled	Hue Enabled	Enables/disables Hue	True False
	Hue Level Auto	Controls the mode for automatic hue adjustment	Off Once Continuous
Saturation	Saturation (percent)	Saturation of the image in percent	
	Saturation Enabled	Enables/disables saturation	True False
	Saturation Auto	Controls the mode for automatic saturation adjustment	Off Once Continuous

8.8 Brightness

Brightness, also known as offset or black level, controls the level of black in an image.

The camera supports brightness control.

- GenICam—[Analog Control](#)
- FlyCapture API—[Setting Brightness Using the FlyCapture API](#)

8.9 Exposure Time

The Blackfly USB 3.1 supports Continuous, Off, and Once control of the image sensor exposure time.

Note: The terms “integration”, “exposure” and “shutter” are interchangeable.

Exposure times are scaled by the divider of the basic frame rate. For example, dividing the frame rate by two (e.g. 15 FPS to 7.5 FPS) causes the maximum exposure time to double (e.g. 66 ms to 133 ms).

The maximum exposure time can be extended beyond the normal range by disabling the frame rate. Once the frame rate is disabled, you should see the maximum value of the exposure time increase.

The supported exposure time range is:

Model	Range	Firmware
BFLY-U3-03S2M-CS	0.023 ms to 31.98 seconds	1.8.3.0
BFLY-U3-03S2C-CS	0.023 ms to 31.98 seconds	1.8.3.0
BFLY-U3-05S2M-CS	0.031 ms to 31.9 seconds	1.4.3.1
BFLY-U3-05S2C-CS	0.031 ms to 31.9 seconds	1.4.3.1
BFLY-U3-13S2M-CS	0.034 ms to 31.9 seconds	1.1.3.0
BFLY-U3-13S2C-CS	0.034 ms to 31.9 seconds	1.1.3.0
BFLY-U3-20S4M-CS	0.072 ms to 31.9 seconds	1.6.3.0
BFLY-U3-20S4C-CS	0.072 ms to 31.9 seconds	1.6.3.0
BFLY-U3-23S6M-C	0.019 ms to 3.9 seconds	1.8.3.0
BFLY-U3-23S6C-C	0.019 ms to 3.9 seconds	1.8.3.0
BFLY-U3-50H5M-C	0.082 ms to 32 seconds	1.8.3.0
BFLY-U3-50H5C-C	0.082 ms to 32 seconds	1.8.3.0

Note: The exposure range may vary depending on the video mode. For more information, see [Video Modes Overview](#).

The time between the end of exposure for consecutive frames is always constant. However, if the exposure time is continually changing (e.g. being controlled by Auto Exposure), the time between the beginning of consecutive integrations will change. If the exposure time is constant, the time between integrations will also be constant.

The camera continually exposes and reads image data off of the sensor under the following conditions:

1. The camera is powered up; and
2. The camera is in free running, not asynchronous trigger, mode. When in trigger mode, the camera simply clears the sensor and does not read the data off the sensor.

The camera continues to expose images even when data transfer is disabled and images are not being streamed to the computer. The camera continues exposing images in order to keep things such as the auto exposure algorithm (if enabled) running. This ensures that when a user starts requesting images, the first image received is properly exposed.

When operating in free-running mode, changes to the exposure value take effect with the next captured image, or the one after next. Changes to exposure in asynchronous trigger mode generally take effect on the next trigger.

To adjust exposure:

- GenICam—[Acquisition Control](#)
- FlyCapture API—[Setting Shutter Using the FlyCapture API](#)

To enable extended exposure:

- FlyCapture SDK example program—[ExtendedShutterEx](#)

8.10 Gain

Gain is the amount of amplification that is applied to a pixel by the A/D converter. An increase in gain can result in a brighter image but also an increase in noise.

The Blackfly USB 3.1 supports Continuous and Once gain modes. The A/D converter provides a PxGA gain stage (white balance/preamp) and VGA gain stage. The main VGA gain stage is available to the user, and is variable between models per the table below.

Model	Range	Firmware
BFLY-U3-03S2M-CS	-10.104 dB to 23.991 dB	1.8.3.0
BFLY-U3-03S2C-CS	-10.104 dB to 23.991 dB	1.8.3.0
BFLY-U3-05S2M-CS	0 dB to 23.991 dB	1.4.3.1
BFLY-U3-05S2C-CS	0 dB to 23.991 dB	1.4.3.1
BFLY-U3-13S2M-CS	-11 dB to 23.99 dB	1.1.3.0
BFLY-U3-13S2C-CS	-11 dB to 23.99 dB	1.1.3.0
BFLY-U3-20S4M-CS	-10.319 dB to 23.991 dB	1.6.3.0

Model	Range	Firmware
BFLY-U3-20S4C-CS	-10.319 dB to 23.991 dB	1.6.3.0
BFLY-U3-23S6M-C	0 dB to 29.996 dB	1.8.3.0
BFLY-U3-23S6C-C	0 dB to 29.996 dB	1.8.3.0
BFLY-U3-50H5M-C	-6.234 dB to 23.991 dB	1.8.3.0
BFLY-U3-50H5C-C	-6.234 dB to 23.991 dB	1.8.3.0

Note: Increasing gain also increases image noise, which can affect image quality. To increase image intensity, try adjusting the lens aperture (iris) and [Exposure Time](#) time first.

To adjust gain:

- GenICam—[Analog Control](#)
 - To access negative Gain values—From Remove Parameter Limits, select Gain and set Remove Limits to True.
- FlyCapture API—[Setting Gain Using the FlyCapture API](#)

8.11 Auto Exposure

Auto exposure allows the camera to automatically control exposure and/or gain in order to achieve a specific average image intensity. Additionally, users can specify the range of allowed values used by the auto-exposure algorithm by setting the auto exposure range, the auto shutter range, and the auto gain range.

Auto Exposure allows the user to control the camera system's automatic exposure algorithm. It has three useful states:

State	Description
Off	Control of the exposure is achieved via setting both Exposure and Gain. This mode is achieved by setting Auto Exposure to Off, or by setting Exposure and Gain to Off.
Manual Exposure Control	The camera automatically modifies Exposure and Gain to try to match the average image intensity to the Auto Exposure value. This mode is achieved by setting Auto Exposure to Off and either/both of Exposure and Gain to Continuous.
Auto Exposure Control	The camera automatically modifies the value in order to produce an image that is visually pleasing. This mode is achieved by setting the all three of Auto Exposure, Exposure, and Gain to Continuous. In this mode, the value reflects the average image intensity.

Auto Exposure can only control the exposure when Exposure and/or Gain are set to Continuous. If only one of the settings is in "auto" mode then the auto exposure controller attempts to control the image intensity using just that one setting. If both of these settings are in "auto" mode the auto exposure controller uses a exposure-before-gain heuristic to try and maximize the signal-to-noise ratio by favoring a longer exposure time over a larger gain value.

The auto exposure algorithm is only applied to the active region of interest, and not the entire array of active pixels.

There are four parameters that affect Auto Exposure:

Auto Exposure Range—Allows the user to specify the range of allowed exposure values to be used by the automatic exposure controller when in auto mode.

Auto Shutter Range—Allows the user to specify the range of exposure values to be used by the automatic exposure controller which is generally some subset of the entire exposure range.

Auto Gain Range—Allows the user to specify the range of gain values to be used by the automatic exposure controller which is generally some subset of the entire gain range.

Auto Exposure ROI—Allows the user to specify a region of interest within the full image to be used for both auto exposure and white balance. The ROI position and size are relative to the transmitted image. If the request ROI is of zero width or height, the entire image is used. Auto Exposure ROI is not supported on quad tap sensor cameras.

To control auto exposure:

- GenICam—[Acquisition Control](#)
- FlyCapture API—[Setting Auto Exposure Using the FlyCapture API](#)

8.12 Sharpness

The Blackfly USB 3.1 supports sharpness adjustment, which refers to the filtering of an image to reduce blurring at image edges. Sharpness is implemented as an average upon a 3x3 block of pixels, and is only applied to the green component of the Bayer tiled pattern. For sharpness values greater than 1024, the pixel is sharpened; for values less than 1024 it is blurred. When sharpness is in auto mode and gain is low, then a small amount of sharpening is applied, which increases as gain decreases. If gain is high, a small amount of blur is applied, increasing as gain increases.

When the camera is outputting raw Bayer data, Sharpness is disabled by default. Otherwise, the default setting is enabled.

To adjust sharpness use:

- GenICam—[Analog Control](#)
- FlyCapture API—[Setting Sharpness Using the FlyCapture API](#)

8.13 Gamma and Lookup Table

The camera supports gamma and lookup table (LUT) functionality. In a GenICam application, gamma and LUT are available when the Image Format Control has On Board Color Processing enabled.

Sensor manufacturers strive to make the transfer characteristics of sensors inherently linear, which means that as the number of photons hitting the imaging sensor increases, the resulting image intensity increases are linear. Gamma can be used to apply a non-linear mapping of the images produced by the camera. Gamma is applied after analog-to-digital conversion and is available in all pixel formats except Raw. Gamma values between 0.5 and 1 result in decreased brightness effect, while values between 1 and 4 produce an increased brightness effect. By default, Gamma is enabled and has a value of 1.25. To obtain a linear response, disable gamma.

For 8-bit, gamma is applied as:

$$\text{OUT} = 255 * (\text{IN}/255)^{1/\text{gamma}}$$

Note: When Gamma is turned on, Lookup Table is turned off. When Lookup Table is turned on, Gamma is turned off.

Lookup Table allows the user to access and control a lookup table (LUT), with entries stored on-board the camera. The LUT is modified under the following circumstances:

- Camera reinitialization
- Changing the current video mode or current video format
- Changing gamma

The LUT can define 2 banks where each bank contains 1 channel. A channel defines a table with a length of $2^{\text{Input_Depth}}$ entries where each entry is *Output_Depth* bits wide. Channel table entries are padded to 32-bits.

Each bank may be read only, write only or both read and write capable as shown by the *LUT_Bank_Rd_Inq* and *LUT_Bank_Wr_Inq* fields. The active bank is set by writing to the *Active_Bank* field of the *LUT_Ctrl* register.

The *Bank_X_Offset_Inq* register gives the offset to start address of the array of channel tables in each bank. Multiple channels can be used to process color video pixel data.

Lookup Table Data Structure

Each bank of channels is composed of entries padded to a complete 32-bits. Each bank is organized as show in the table below.

Cn: Channel Number

En : Entry Number

C(0)E(0)
...
...
C(0)E($2^{\text{Input_Depth}}-1$)
C(1)E(0)
...
...
C(1)E($2^{\text{Input_Depth}}-1$)
...
...
...
C(Number_of_Channels-1)E(0)
...
...
C(Number_of_Channels-1) E($2^{\text{Input_Depth}}-1$)

Related Knowledge Base Articles

Title	Article
How is gamma calculated and applied?	Knowledge Base Article 10282

To adjust Lookup Table (LUT):

- GenICam—[GenICam Features](#)

To adjust gamma:

- GenICam—[Analog Control](#)
- FlyCapture API—[Setting Gamma Using the FlyCapture API](#)

8.14 High Dynamic Range (HDR) Imaging

Generally speaking, digital camera systems are not capable of accurately capturing many of the high dynamic range scenes that they are exposed to in real world settings. That is, they may not be able to capture features in both the darkest and brightest areas of an image simultaneously - darker portions of the image are too dark or brighter portions of the image are too bright. High Dynamic Range (HDR) mode helps to overcome this problem by capturing images with varying exposure settings. HDR is best suited for stationary applications.

The camera can be set into an HDR mode in which it cycles between 4 user-defined shutter and gain settings, applying one gain and shutter value pair per frame. This allows images representing a wide range of shutter and gain settings to be collected in a short time to be combined into a final HDR image later. The camera does not create the final HDR image; this must be done by the user.

The HDR interface contains gain and shutter controls for 4 consecutive frames. When **Enable high dynamic range** is checked, the camera cycles between settings 1-4, one set of settings per consecutive frame.

Note: HDR is supported on the BFLY-U3-23S6 model only.

To enable HDR:

- FlyCapture SDK example program—[HighDynamicRangeEx](#)

8.15 Image Flip/Mirror

The camera supports horizontal image mirroring.

To enable image mirroring use:

- GenICam—[Image Format Control](#)

8.16 Embedded Image Information

This setting controls the frame-specific information that is embedded into the first several pixels of the image. The first byte of embedded image data starts at pixel 0,0 (column 0, row 0) and continues in the first row of the image data: (1,0), (2,0), and so forth. Users using color cameras that perform Bayer color processing on the computer must extract the value from the non-color processed image in order for the data to be valid.

Note: Embedded image values are those in effect at the end of shutter integration.

Each piece of information takes up 32-bits (4 bytes) of the image. When the camera is using an 8-bit pixel format, this is 4 pixels worth of data.

The following frame-specific information can be provided:

- Timestamp
- Gain
- Exposure
- Brightness
- White Balance
- Frame counter
- Strobe Pattern counter
- GPIO pin state
- ROI position

If you turned on all possible options the first 40 bytes of image data would contain camera information in the following format, when accessed using the FlyCapture 2 API:

(assuming `unsigned char* data = rawImage.GetData();` and an Image object `rawImage`):

- `data[0]` = first byte of Timestamp data
- `data[4]` = first byte of Gain data
- `data[24]` = first byte of Frame Counter data

If only Exposure embedding were enabled, then the first 4 bytes of the image would contain Exposure information for that image. Similarly, if only Brightness embedding were enabled, the first 4 bytes would contain Brightness information.

For monochrome cameras, white balance is still included, but no valid data is provided.

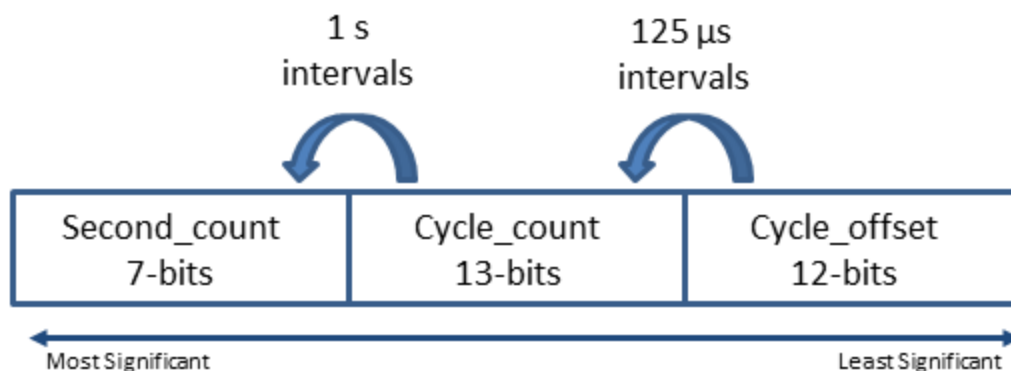
To access embedded information:

- CSRs—[FRAME_INFO: 12F8h](#)

Interpreting Timestamp information

The CYCLE_TIME register is located at 1EA8h.

The Timestamp format is as follows (some cameras replace the bottom 4 bits of the cycle offset with a 4-bit version of the Frame Counter):



Cycle_offset increments from 0 to x depending on implementation, where x equals one cycle_count.

Cycle_count increments from 0 to 7999, which equals one second.

Second_count increments from 0 to 127.

All counters reset to 0 at the end of each cycle.

Note: On USB devices, the four least significant bits of the timestamp do not accurately reflect the cycle_offset and should be discounted.

Interpreting ROI information

The first two bytes are the distance from the left frame border that the region of interest (ROI) is shifted. The next two bytes are the distance from the top frame border that the ROI is shifted.

8.17 White Balance

White balance is applicable to color models only.

The Blackfly USB 3.1 supports white balance adjustment, which is a system of color correction to account for differing lighting conditions. Adjusting white balance by modifying the relative gain of R, G and B in an image enables white areas to look "whiter". Taking some subset of the target image and looking at the relative red to green and blue to green response, the objective is to scale the red and blue channels so that the response is 1:1:1.

The user can adjust the red and blue values. Both values specify relative gain, with a value that is half the maximum value being a relative gain of zero.

White Balance has two states:

State	Description
Off	The same gain is applied to all pixels in the Bayer tiling.
On/Manual	The Red value is applied to the red pixels of the Bayer tiling and the Blue value is applied to the blue pixels of the Bayer tiling.

The following table illustrates the default gain settings for most cameras.

	Red	Blue
Black and White	32	32
Color	1023	1023

The camera can also implement Continuous and Once white balance. One use of Continuous and Once white balance is to obtain a similar color balance between cameras that are slightly different from each other. In theory, if different cameras are pointed at the same scene, using Continuous and Once results in a similar color balance between the cameras.

Once only attempts to automatically adjust white balance for a set period of time before stopping. It uses a “white detection” algorithm that looks for “whitish” pixels in the raw Bayer image data. Once adjusts the white balance for a specific number of iterations; if it cannot locate any whitish pixels, it will gradually look at the whitest objects in the scene and try to work off them. It will continue this until has completed its finite set of iterations.

Continuous is continually adjusting white balance. It differs from Once in that it works almost solely off the whitest objects in the scene.

Note: The white balance of the camera before using Continuous and Once must already be relatively close; that is, if Red is set to 0 and Blue is at maximum (two extremes), Continuous and Once will not function as expected. However, if the camera is already close to being color balanced, then Continuous and Once will function properly.

Note: The term Continuous is the same as Auto and the term Once is the same as One Push.

To adjust white balance:

- GenICam—[Analog Control](#)
- FlyCapture API—[Setting White Balance Using the FlyCapture API](#)

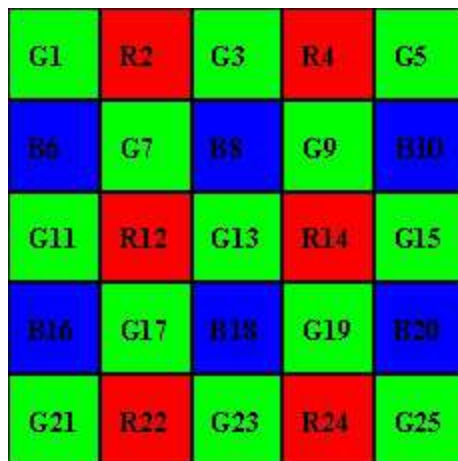
8.18 Bayer Color Processing

Bayer color processing is applicable to color models only.

A Bayer tile pattern color filter array captures the intensity red, green or blue in each pixel on the sensor. The image below is an example of a Bayer tile pattern.

To determine the actual pattern on your camera, query the Pixel Color Filter GenICam feature.

Figure 8.2: Example Bayer Tile Pattern



In order to produce color (e.g. RGB, YUV) and greyscale (e.g. Y8, Y16) images, color models perform on-board processing of the Bayer tile pattern output produced by the sensor.

Conversion from RGB to YUV uses the following formula:

To convert the Bayer tile pattern to greyscale, the camera adds the value for each of the RGB components in the color processed pixel to produce a single greyscale (Y) value for that pixel, as follows:

To control Bayer color processing:

- GenICam—[Image Format Control](#)
- FlyCapture API—[Accessing Raw Bayer Data using FlyCapture](#)

Accessing Raw Bayer Data

Users interested in accessing the raw Bayer data to apply their own color conversion algorithm or one of the SDK library algorithms should acquire images using a video mode that supports Raw pixel encoding.

The actual physical arrangement of the red, green and blue "pixels" for a given camera is determined by the arrangement of the color filter array on the imaging sensor itself. The format, or order, in which this raw color data is streamed out, however, depends on the specific camera model and firmware version.

Related Knowledge Base Articles

Title	Article
Different color processing algorithms	Knowledge Base Article 10141
Writing color processing software and color interpolation algorithms	Knowledge Base Article 10144
How is color processing performed on my camera's images?	Knowledge Base Article 10318

8.19 Hue

Hue is applicable to color models only.

This provides a mechanism to control the Hue component of the images being produced by the Blackfly USB 3.1, given a standard Hue, Saturation, Value (HSV) color space.

To adjust hue use:

- GenICam—[Analog Control](#)
- FlyCapture API—[Setting Hue Using the FlyCapture API](#)

8.20 Saturation

Saturation is applicable to color models only.

This provides a mechanism to control the Saturation component of the images being produced by the Blackfly USB 3.1, given a standard Hue, Saturation, Value (HSV) color space.

Note: Saturation in this context does not refer to the saturation of a sensor charge.

To adjust saturation use:

- GenICam—[Analog Control](#)
- FlyCapture API—[Setting Saturation Using the FlyCapture API](#)

9 Troubleshooting

9.1 Downloads and Support

FLIR endeavors to provide the highest level of technical support possible to our customers. Most support resources can be accessed through the [Support](#) section of our website.

The first step in accessing our technical support resources is to obtain a Customer Login Account. This requires a valid name and email address. To apply for a Customer Login Account go to our [Downloads](#) page.

Customers with a Customer Login Account can access the latest **software** and **firmware** for their cameras from our website. We encourage our customers to keep their software and firmware up-to-date by downloading and installing the latest versions.

9.1.1 Finding Information

FlyCapture SDK—The FlyCapture SDK provides API examples and the FlyCap camera evaluation application. Available from our [Downloads](#) page.

API Documentation—The installation of the FlyCapture SDK comes with API references for C++, C#, and C code. Available from Start Menu→All Programs→Point Grey FlyCapture2 SDK→Documentation

Product Documentation—The camera's *Getting Started Manual* provides information on installing components and software needed to run the camera. The *Technical Reference* provides information on the camera's specifications, features and operations, as well as imaging and acquisition controls. They are available from the [Downloads](#) page.

Knowledge Base—A database of articles and application notes with answers to common questions as well as articles and tutorials about hardware and software systems. Available from our [Knowledge Base](#).

Learning Center—Our [Learning Center](#) contains links to many resources including videos, case studies, popular topics, other application notes, and information on sensor technology.

9.1.2 Contacting Technical Support

Before contacting Technical Support, have you:

1. Read the product documentation?
2. Searched the Knowledge Base?
3. Downloaded and installed the latest version of software and/or firmware?

If you have done all the above and still can't find an answer to your question, contact our [Technical Support](#) team.

9.2 Camera Diagnostics

Use the following parameters to monitor the error status of the camera and troubleshoot problems:

Time from Initialize—This reports the time, in seconds, since the camera was initialized during a hard power-up. This is different from powering up the camera, which will not reset this time.

Time from Bus Reset—This reports the time, in seconds, since the last bus reset occurred. This will be equal to the Time from Initialize if no reset has occurred since the last time the camera was initialized.

Transmit Failure—This contains a count of the number of failed frame transmissions that have occurred since the last reset. An error occurs if the camera cannot arbitrate for the bus to transmit image data and the image data FIFO overflows.

Camera Log—This provides access to the camera's 256 byte internal message log, which is often useful for debugging camera problems. Contact [technical support](#) for interpretation of message logs.

To access the camera diagnostics

- GenICam—[Device Control](#)
- CSRs—[Control and Status Registers](#)

9.3 Status Indicator LED

The user can turn off the camera's status LED. LEDs are re-enabled the next time the camera is power cycled.

LED Status	Description
Off	Not receiving power
Steady green	Receiving power
Flashing yellow/Steady yellow	Initializing FPGA
Steady yellow-green	Sensor powered down Insufficient power
Steady bright green	Acquiring and transmitting images
Flashing bright, then brighter green	Accessing camera registers (no image acquisition)
Flashing green and red	Updating firmware
Flashing red	Temporary problem
Steady red	Serious problem

For information on the LED register, see [LED_CTRL: 1A14h on page 112](#).

9.4 Test Pattern

The camera is capable of outputting continuous static images for testing and development purposes. The test pattern image is inserted into the imaging pipeline immediately prior to the transfer to the on-board FIFO, and is therefore not subject to changes in imaging parameters.

Note: If you enable raw Bayer output when operating in a monochrome data format it produces an image shift effect in the test pattern.

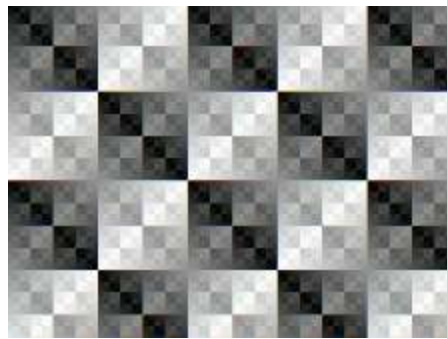


Figure 9.1: Test Pattern Sample Image

- GenICam—[Image Format Control](#)

9.5 Channel Balancing

Some camera sensors are capable of running in a multiple output (or "multi tap") mode. In multiple output mode, the sensor is capable of reading out data at very high speed. This allows the camera to operate at fast frame rates.

In single output mode, all pixels are shifted off the sensor to the lower left corner of the sensor. In a multiple output mode the image is divided into sections for reading off the sensor. For example, in a dual output mode, the right half of the horizontal CCD is reversed and is read off the sensor at the lower right, while the left half is still read off at the lower left.

As a result of pixel data coming off the sensor at different locations, multiple analog-to-digital (A/D) converters are required to convert the electrical charge to digital output. All A/D converters, even those of the same make/model, will have subtle differences in the way they process the same input information. This can result in different output data given the same input and same A/D conversion parameters. Specifically, this can result in the difference in image intensities between the different sections of an image.

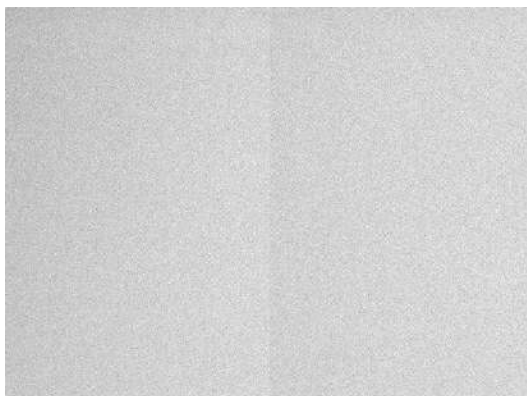


Figure 9.2: Example of dual channel image with no balancing

To address this issue, FLIR "balances" every multiple tap unit as part of the quality control process. This balancing process attempts to minimize the difference in gains that result from the different A/D converters.

Note: Some slight differences may still be visible between 0-10 dB.

Note: Balancing is only done in full resolution modes.

9.6 Link Recovery

Link recovery occurs when communication between the camera and the host controller has encountered enough errors to require a renegotiation.

If you encounter link recover errors try the following, in order:

1. Ensure the cable is seated correctly in the camera and in the host controller.
2. Unplug and reconnect the cable.
3. Use a different cable.
4. Update the camera firmware to the latest version (available on the [Downloads page](#)).
5. Update the SDK to the latest version (available on the [Downloads page](#)).
6. Simplify the topology of the system by using shorter cables, or by not chaining cables.
7. Use a different host controller, even one by the same manufacturer.

If none of these options helps to remove the link recovery errors, contact support.

Note: The FlyCap program provides diagnostics with an indication of the Link Recovery Count. This is viewable from the Diagnostics tree in the left pane of the streaming window, or by clicking the lower left corner of the streaming window to see Event Statistics.

9.7 Blemish Pixel Artifacts

Cosmic radiation may cause random pixels to generate a permanently high charge, resulting in a permanently lit, or 'glowing,' appearance. FLIR tests for and programs white blemish pixel correction into the camera firmware.

In very rare cases, one or more pixels in the sensor array may stop responding and appear black (dead) or white (hot/stuck).

9.7.1 Pixel Defect Correction

FLIR tests for blemish pixels on each camera. The mechanism to correct blemish pixels is hard-coded into the camera firmware, and can be turned off and on by the user. Pixel correction is on by default. The correction algorithm involves applying the average color or grayscale values of neighboring pixels to the blemish pixel.

Note: Pixel correction is not done in any of the [binning modes](#).

Related Knowledge Base Articles

Title	Article
How we test for white blemish pixels	Technical Application Note 10561

9.8 Vertical Smear Artifact

When a strong light source is shone on the camera, a faint bright line may be seen extending vertically through an image from a light-saturated spot. Vertical smear is a byproduct of the interline transfer system that extracts data from the CCD.

Smear is caused by scattered photons leaking into the shielded vertical shift register. When the pixel cells are full, some charges may spill out in to the vertical shift register. As the charge shifts in/out of the light sensitive sensor area and travels down the vertical shift register, it picks up the extra photons and causes a bright line in the image.

Smear above the bright spot is collected during read out while smear below the bright spot is collected during read in.

9.8.1 Smear Reduction

Smear may be minimized using one or more of the following techniques:

- Reduce the bright light source.
- Increase the shutter time/lower the frame rate. This increases the amount of time light is collected in the photosensors relative to the time in the vertical transfer register.
- Turn the light source off before and after exposure by using a mechanical or LCD shutter.
- Use a pulsed or flashed light source. A pulsed light of 1/10,000 duration is sufficient in most cases to allow an extremely short 100 ns exposure without smear.
- Increase light collimation by using a lens with variable aperture. Note that an effect of closing the iris is a darker image.
- Use a [low smear trigger mode](#) which may reduce the effect of smear. This trigger mode may not be available on all models.

Related Knowledge Base Articles

Title	Article
Vertical bleeding or smearing from a saturated portion of an image	Knowledge Base Article 10317

A FlyCapture API Code Samples

A.1 Setting a GPIO Pin to Strobe Using the FlyCapture API

The following FlyCapture code sample uses the C++ interface to do the following:

- Configures GPIO1 as the strobe output pin.
- Enables strobe output.
- Specifies an active high (rising edge) strobe signal.
- Specifies that the strobe signal begin 1 ms after the shutter opens.
- Specifies the duration of the strobe as 1.5 ms.

Assuming a `Camera` object `cam`:

```
StrobeControl mStrobe;  
mStrobe.source = 1;  
mStrobe.onOff = true;  
mStrobe.polarity = 1;  
mStrobe.delay = 1.0f;  
mStrobe.duration = 1.5f  
cam.SetStrobe(&mStrobe);
```

A.2 Setting a Standard Video Mode, Format and Frame Rate Using the FlyCapture API

The following FlyCapture code snippet sets the camera to: 640x480 Y8 at 60 FPS.

```
Camera.SetVideoModeandFrameRate( VIDEOMODE_640x480Y8 , FRAMERATE_60 );
```

A.3 Asynchronous Hardware Triggering Using the FlyCapture API

The following FlyCapture code sample uses the C++ interface to do the following:

- Sets the trigger mode to Trigger Mode 0.
- Configures GPIO0 as the trigger input source.
- Enables triggered acquisition.
- Specifies the trigger signal polarity as an active high (rising edge) signal.

Assuming a `Camera` object `cam`:

```
TriggerMode mTrigger;  
mTrigger.mode = 0;  
mTrigger.source = 0;  
mTrigger.parameter = 0;  
mTrigger.onOff = true;  
mTrigger.polarity = 1;  
cam.SetTriggerMode(&mTrigger);
```

A.4 Setting Brightness Using the FlyCapture API

The following FlyCapture code snippet adjusts brightness to 0.5% using the C++ interface. The snippet assumes a `Camera` object `cam`.

```
//Declare a Property struct.  
Property prop;  
//Define the property to adjust.  
prop.type = BRIGHTNESS;  
//Ensure the property is set up to use absolute value control.  
prop.absControl = true;  
//Set the absolute value of brightness to 0.5%.  
prop.absValue = 0.5;  
//Set the property.  
error = cam.SetProperty( &prop );
```

A.5 Setting Shutter Using the FlyCapture API

The following FlyCapture code snippet adjusts the shutter speed to 20 ms using the C++ interface. The snippet assumes a `Camera` object `cam`.

```
//Declare a Property struct.  
Property prop;  
//Define the property to adjust.  
prop.type = SHUTTER;
```

```
//Ensure the property is on.
prop.onOff = true;
//Ensure auto-adjust mode is off.
prop.autoManualMode = false;
//Ensure the property is set up to use absolute value control.
prop.absControl = true;
//Set the absolute value of shutter to 20 ms.
prop.absValue = 20;
//Set the property.
error = cam.SetProperty( &prop );
```

A.6 Setting Gain Using the FlyCapture API

The following FlyCapture code snippet adjusts gain to 10.5 dB using the C++ interface, and assumes a `Camera` object `cam`.

```
//Declare a Property struct.
Property prop;
//Define the property to adjust.
prop.type = GAIN;
//Ensure auto-adjust mode is off.
prop.autoManualMode = false;
//Ensure the property is set up to use absolute value control.
prop.absControl = true;
//Set the absolute value of gain to 10.5 dB.
prop.absValue = 10.5;
//Set the property.
error = cam.SetProperty( &prop );
```

A.7 Setting Auto Exposure Using the FlyCapture API

The following FlyCapture code snippet adjusts auto exposure to -3.5 EV using the C++ interface. The snippet assumes a `Camera` object `cam`.

```
//Declare a Property struct.  
Property prop;  
//Define the property to adjust.  
prop.type = AUTO_EXPOSURE;  
//Ensure the property is on.  
prop.onOff = true;  
//Ensure auto-adjust mode is off.  
prop.autoManualMode = false;  
//Ensure the property is set up to use absolute value control.  
prop.absControl = true;  
//Set the absolute value of auto exposure to -3.5 EV.  
prop.absValue = -3.5;  
//Set the property.  
error = cam.SetProperty( &prop );
```

A.8 Setting Sharpness Using the FlyCapture API

The following FlyCapture code snippet adjusts sharpness to 1500 using the C++ interface. The snippet assumes a `Camera` object `cam`.

```
//Declare a Property struct.  
Property prop;  
//Define the property to adjust.  
prop.type = SHARPNESS;  
//Ensure the property is on.  
prop.onOff = true;  
//Ensure auto-adjust mode is off.  
prop.autoManualMode = false;  
//Set the value of sharpness to 1500.  
prop.valueA = 1500;  
//Set the property.  
error = cam.SetProperty( &prop );
```

A.9 Setting Gamma Using the FlyCapture API

The following FlyCapture code snippet adjusts gamma to 1.5 using the C++ interface. The snippet assumes a `Camera` object `cam`.

```
//Declare a Property struct.  
Property prop;  
//Define the property to adjust.  
prop.type = GAMMA;  
//Ensure the property is on.  
prop.onOff = true;  
//Ensure the property is set up to use absolute value control.  
prop.absControl = true;  
//Set the absolute value of gamma to 1.5  
prop.absValue = 1.5;  
//Set the property.  
error = cam.SetProperty( &prop );
```

A.10 Setting White Balance Using the FlyCapture API

The following FlyCapture code snippet adjusts the white balance red channel to 500 and the blue channel to 850 using the C++ interface. The snippet assumes a `Camera` object `cam`.

```
//Declare a Property struct.  
Property prop;  
//Define the property to adjust.  
prop.type = WHITE_BALANCE;  
//Ensure the property is on.  
prop.onOff = true;  
//Ensure auto-adjust mode is off.  
prop.autoManualMode = false;  
//Set the white balance red channel to 500.  
prop.valueA = 500;  
//Set the white balance blue channel to 850.
```

```
prop.valueB = 850;
//Set the property.
error = cam.SetProperty( &prop );
```

A.11 Accessing Raw Bayer Data using FlyCapture

Using the FlyCapture SDK, raw image data can be accessed programmatically via the `getData` method of the `Image` class. In Raw8 modes, the first byte represents the pixel at [row 0, column 0], the second byte at [row 0, column 1], and so on.

Read the `BAYER_TILE_MAPPING` register 0x1040 to determine the current Bayer output format (RGGB, GRBG, and so on). Using a Bayer format of RGGB, for example, the `getData` method returns the following (assuming `char* data = rawImage.GetData()`; and an `Image` object `rawImage`):

- `data[0]` = Row 0, Column 0 = red pixel (R)
- `data[1]` = Row 0, Column 1 = green pixel (G)
- `data[640]` = Row 1, Column 0 = green pixel (G)
- `data[641]` = Row 1, Column 1 = blue pixel (B)

A.12 Setting Hue Using the FlyCapture API

The following FlyCapture code snippet adjusts hue to -30 deg. using the C++ interface. The snippet assumes a `Camera` object `cam`.

```
//Declare a Property struct.
Property prop;
//Define the property to adjust.
prop.type = HUE;
//Ensure the property is on.
prop.onOff = true;
//Ensure the property is set up to use absolute value control.
prop.absControl = true;
//Set the absolute value of hue to -30 deg.
prop.absValue = -30;
//Set the property.
error = cam.SetProperty( &prop );
```

A.13 Setting Saturation Using the FlyCapture API

The following FlyCapture code snippet adjusts saturation to 200% using the C++ interface. The snippet assumes a Camera object `cam`.

```
//Declare a property struct.  
Property prop;  
//Define the property to adjust.  
prop.type = SATURATION;  
//Ensure the property is on.  
prop.onOff = true;  
//Ensure auto-adjust mode is off.  
prop.autoManualMode = false;  
//Ensure the property is set up to use absolute value control.  
prop.absControl = true;  
//Set the absolute value of saturation to 200%.  
prop.absValue = 200;  
//Set the property.  
error = cam.SetProperty( &prop );
```

B FlyCapture SDK Examples

The FlyCapture SDK includes a number of examples in C, C++, C#, and VB.NET to help get you started in some basic camera programming tasks. Some examples may not be supported by your camera model.

Note: The full example source code can be found in the \src directory of the FlyCapture2 SDK installation. To access the examples workspace from the Start menu, select **Program Files>Point Grey FlyCapture2 SDK >Examples**

B.1 AsyncTriggerEx

The AsyncTriggerEx example program demonstrates some of the basic asynchronous trigger capabilities of compatible FLIR machine vision cameras.

This program only works with cameras that can be asynchronously triggered, either using an external hardware trigger or by using the camera's internal software trigger.

The camera is started and put into trigger mode. The user can then either press a key to software trigger the camera, or trigger through an external hardware trigger. The example captures a specified number of images and then exits.

Available for:

- C++
- C#
- VB.NET

B.2 BusEventsEx

The BusEventsEx demonstrates how to Register for Bus Events such as Camera Arrival/Removal and Bus Resets using the managed API.

Available for:

- VB.NET

B.3 CustomImageEx

The CustomImageEx example program demonstrates how to configure a FLIR machine vision camera to output custom sized images - the FlyCapture equivalent of the IIDC specifications 'Format 7'. Custom image modes are often useful for achieving faster frame rates, reducing the resolution of an image, and allowing more cameras to run on a single bus by reducing bandwidth requirements.

The program creates a context and initializes the first camera on the 1394 bus. It then queries the camera to determine the custom image modes, resolution sizes, unit sizes and pixel formats the camera supports. The information returned by QueryFormat7Info() is the same kind of information you would see in FlyCap using the Format7 tab.

The program then starts the camera in custom image mode using parameters defined at the beginning of the code. Calling `SetFormat7Configuration()` with these parameters is essentially the same thing as setting these parameters in FlyCap and clicking "Apply". A number of images are grabbed in this custom image mode. The final image is then color-processed and saved in .bmp format to disk.

Available for:

- C++
- C#
- VB.NET

B.4 ExtendedShutterEx

The ExtendedShutterEx example program demonstrates how to enable and calculate extended integration times for your camera. The way this is done can differ between cameras.

Many applications require extended shutter (integration) times up to several seconds long. Most FLIR machine vision cameras implement extended shutter functionality in one of two ways:

1. By turning off the FRAME_RATE register 0x83C. This effectively stops the camera from transmitting images at fixed frame intervals; the frame rate becomes dependent on the shutter time.
2. By enabling extended shutter via the EXTENDED_SHUTTER register 0x1028.

The program begins by initializing the first camera on the bus and uses `GetProperty()` to determine if it implements the FRAME_RATE register. If it does, it turns the frame rate off. If the camera does not implement this register, the program then checks to see if the camera implements the EXTENDED_SHUTTER register. If it does, it accesses this register to put the camera into extended shutter mode. Otherwise, the user is notified that the camera does not implement extended shutter and the program exits.

Once the camera is in extended shutter mode, it is started in the default mode and frame rate. A series of images are grabbed, and their timestamps printed as a way of verifying that the extended shutter is working.

B.5 FlyCap2CameraControl

The FlyCap2CameraControl is the source code to our main Control dialog that can be launched through the Flycapture2GUI API. This source demonstrates all possible controls available in FlyCapture2 SDK.

Available for:

- C#

B.6 FlyCap2_GTKmm

This example allows a user to select a camera to start, and then starts streaming images to screen. It is written using C++ with the GTKmm graphical framework. There are options to modify camera settings and display a histogram window. Images can be saved as a single image capture or multiple sequential images of various formats. Data can be saved by specifying the number of frames to capture, the length of time, or an indefinite stream where the user selects when to start and stop the camera recording.

A single instance of FlyCap2_GTKmm can only run one camera. However, multiple applications can be run to view different cameras. Simply select more than one camera when starting FlyCap2_GTKmm. Note that the number of active cameras is limited by the bandwidth of the bus.

Note: Beginning with FlyCapture version 2.2, the GTK Runtime libraries required to run this example are no longer pre-installed. To download the GTK Runtime, go to our [downloads site](#).

B.7 FlyCap2MFC

The FlyCap2MFC example is the equivalent of the FlyCap2 example program, except it uses the Microsoft Foundation Class Library to implement the graphical user interface. Like FlyCap2, it is the main FLIR machine vision camera application used to work with single lens cameras. It allows a user to select a camera to start, and then starts streaming the images to screen. There are options to modify camera settings and save single images.

Note: Visual Studio Standard Edition is required to build the FlyCap2MFC demo application. Express Edition does not include the MFC library.

While a single instance of FlyCap2MFC can only open one camera, multiple FlyCap2MFC applications can be run to view more than one camera. Note that the number of active cameras is limited by the bandwidth of the bus.

B.8 FlyCapture2GUI

This example contains the same source code that is used for the Camera Selection and Camera Control dialogs in FlyCapture2.

As a result, it uses many of the features available in the FlyCapture2 API and is a useful source for discovering how to perform many common functions, such as camera property manipulation, using the FlyCapture2 API.

Available for:

- C++
- C#

B.9 FlyCapture2SimpleGUI_WPF

The FlyCapture2SimpleGUI_WPF shows how to build the WPF GUI example.

Available for:

- C#

B.10 FlyCapture2Test

The FlyCapture2Test example program is a simple program designed to report information related to all compatible cameras attached to the host system, capture a series of images from a single camera, record the amount of time taken to grab these images, then save the last image in the current directory.

Available for:

- C
- C++
- C#
- VB.NET

B.11 GigEGrabEx

The GigEGrabEx example program demonstrates how to use the GigECamera object to set up a GigE Vision specific Image grabbing loop.

Available for:

- C
- C++
- C#
- VB.NET

B.12 GrabCallbackEx

The GrabCallbackEx example program demonstrates how to set up an asynchronous image callback application using FlyCapture2 API.

Available for:

- C#
- VB.NET

B.13 HighDynamicRangeEx

The HighDynamicRangeEx example demonstrates the use of the High Dynamic Range (HDR) functionality. This example can only be used on cameras which support HDR.

When HDR mode is enabled, the shutter and gain settings for each image alternate between four sets of shutter and gain settings in the HDR register. All 4 registers must be used; the images cycle through registers HDR0 to HDR3 and then back to HDR0 again. This cycle continues until HDR mode is turned off.

The example initializes the camera, and verifies that HDR is supported. The four HDR registers are then set up with different, increasing values for shutter and gain.

The next 4 images are then grabbed and saved to disk. The user can look at these images and verify that each image corresponds to the settings for each HDR register.

Once the images are saved to disk, the program cleans up and exits.

B.14 ImageEventEx

Note: This example works with FireWire cameras only.

This example illustrates how users can implement partial image event notification. Partial image event notification is a mechanism that provides the user with access to image data as it arrives in the PC's memory, before the entire image is available.

This functionality is achieved by specifying a number of events, which are tied to various locations in the image. The events are then signalled as the corresponding portion of the image arrives on the PC. This allows the user to start processing the data immediately without having to wait for image transmission to complete. If you specify one event, it occurs at the end of the image. If you specify two events, the first occurs near the beginning of the image, and the second occurs at the end. If you specify more than two events, they are spread evenly among the remainder of the image.

Partial image event notification is particularly useful in applications requiring extremely low latency, such as moving the camera and stopping only to take pictures. In this case, setting two events, with the first occurring near the beginning of the image, is a good method for indicating the end of integration and that it is safe to move the camera without disrupting image capture.

Partial image event notification is also available in custom image mode; however, there are some additional considerations when using this mode. Event notifications must be set on packet boundaries, so you must compute the total image size, including padding, when deciding where to set event sizes. There will be at most one padded packet transmitted, so the ceiling of the computed image size divided by the packet size returns the number of packets transmitted per image:

$$\text{numOfPackets} = \text{ceiling}((\text{rows} * \text{cols} * \text{bytesPerPixel}) / \text{bytesPerPacket})$$

If the camera has already been started with the chosen bytes per packet, this value can be queried from the format 7 registers. See the entry for `PACKET_PER_FRAME_INQ (0x048)` in the *FLIR Machine Vision Camera Register Reference*.

Note:

Partial image event notification operates differently between the Windows and Linux operating systems in the following ways:

- On Windows, if more than one image event is specified, the first event occurs after the PC receives the first packet. The remainder of the events are equally distributed along the length of the image. On Linux, all events are equally distributed along the image. However, if an image is transmitted in more than one packet, there are no notifications after the first packet is transmitted.
- On Linux, synchronizing image transmission on the sy-bit is disabled when using partial image event notification. As a result, in certain cases when the CPU is heavily loaded and the image rendering

software is not cycling for a long period, the image stream may fall out of synch and become corrupted. To re-synchronize transmission, stop and re-start isochronous image grabbing and transmission.

Note: Depending on your operating system, for this example to work on your PC, you may need to install a hotfix from Microsoft.

Related Knowledge Base Articles

Title	Article
Recommended or required Windows Service Packs and Hotfixes	Knowledge Base Article 10038

B.15 MultipleCameraEx

Note: This example works with FireWire cameras only.

This example starts multiple cameras using the StartSyncCapture() function. This function synchronizes image grabbing across all cameras. Additionally, it enables timestamps to be embedded in images, allowing users to obtain the exact timing of each camera's exposure.

B.16 MultipleCameraWriteToDiskEx

The MultipleCameraWriteToDiskEx shows how to write to disk from multiple cameras.

Available for:

- C++

B.17 MultiSyncEx




Note: This example works with FireWire cameras only.

This example synchronizes 1394 cameras on same PC and across PCs. Across PC synchronization is achieved by linking all of the computers to a single 1394 bus dedicated to sharing timing information (the timing bus). This requires that a 1394 card on each machine be dedicated to the timing bus.

Note: This example does not perform image grabbing. You have to write your own image acquisition program or use existing FlyCapture2 examples such as MultipleCameraEx to perform synchronized image grab.

This example lists detected 1394 cameras on current system and displays current synchronization status and time duration since sync was established. Cameras connected to other computers in the sync network cannot be seen from local computer.

Detected cameras are highlighted in following ways to indicate synchronization status:

Color	Taskbar Icon	Meaning
Red		The cameras are not synchronized or there are not enough cameras to synchronize.
Yellow		The cameras are in the process of synchronizing.
Green		The cameras are synchronized

B.18 SaveImageToAviEx

This example saves a series of images to AVI files. The program starts the first camera attached to the host system, opens an AVI file, captures a series of images from the camera, and appends the images to the AVI file.

B.19 SaveImageToFlashEx

The SaveImageToFlashEx utility is a basic example which utilizes the data flash on the camera. When an image is grabbed, it is saved directly to flash instead of sending the data out. Not all cameras support data flash. On supported cameras, flash size varies by model, so the size of the image that can be stored varies. Consult your camera's documentation for more information.

Once the image is stored in the camera, the image can be recovered at any time on any PC.

The example uses a FlashMode enumeration to capture the image **(-c)** or save the stored image to disk **(-r)**.

B.20 SerialPortEx

Note: Blackfly USB 3.1 does not support the Serial Port feature.

This example illustrates how users can transmit and receive characters by using the camera's serial buffer system.

This example creates the camera context and does the following:

- Allocates a GUI handle to be used in all successive calls
- Displays the camera selection dialog

- Initializes the selected camera on the bus and associates it with the given context
- Checks to make sure that the serial port is actually supported
- Creates a thread to receive data and display the data

OnTransmit() is used to transmit data out of the camera's serial port based on user input.

ReceiveLoop() is used to get the connection parameters from the camera, update the dialog, verify the receive buffer status and determine the amount of data to be read, read the data and display the data in the window.

Users can use the 'Write Register Value' button to set the serial port register values, and use 'Read Register Value' to get the serial port register values.

C GenICam Features

The following features are included in the XML device description file on the camera to control, monitor, and query the camera operation. Not all operations can be controlled using the XML file; those not included are controlled via CSRs.

Note: Except where noted, these features conform to the GenICam Standard Features Naming Convention (SFNC) v1.5. The full SFNC can be found on the [EMVA.org website](http://EMVA.org) and contains more details than provided below. **Not all features are available on all cameras.**

Non-SFNC features are noted with a grey background.

C.1 Device Control

Name	Display Name	Description	Value
DeviceVendorName	Vendor Name	Name of the manufacturer	
DeviceModelName	Model Name	Model name of the device	Blackfly USB 3.1 <model number>
DeviceVersion	Device Version	FPGA version	
DeviceSerialNumber	Device Serial Number	Camera Serial Number	
DeviceUserID	Device User ID	User defined name	
DeviceGenCPVerMajor	Device GenCP Version Major	Major version of the GenCP protocol supported by the camera	
DeviceGenCPVerMinor	Device GenCP Version Minor	Minor version of the GenCP protocol supported by the camera	
DeviceFamilyName	Family Name	Family name of the device	Blackfly USB 3.1
Timestamp	Timestamp	Reports the current value of the camera timestamp counter	
TimestampLatch	Timestamp Latch	Latches the current camera time into the timestamp register	Write Only
TimestampIncrement	Timestamp Increment	Indicates the timestamp increment in ns/tick	
SensorDescription	Sensor Description	Sensor description of the camera	
DeviceFirmwareVersion	Device Firmware Version	Firmware version	
DeviceScanType	Device Scan Type	Scan type of the sensor	Areascan (2D sensor)

Name	Display Name	Description	Value
DeviceTemperature	Device Temperature (C)	Device temperature in degrees Celsius	
DeviceReset	Device Reset	Resets the device to its initial state and default settings	Write Only
DeviceUptime	Device Uptime	Time since the camera powered up	
DeviceMaxThroughput	Device Max Throughput	Indicates maximum data in bytes from the camera each second	
DeviceLinkThroughputLimit	Device Link Throughput Limit	User defined bandwidth limitation	
DeviceIndicatorMode	Device Indicator Mode	Controls the behavior of the indicators (such as LEDs) showing the status of the Device	Inactive: Device's indicators are inactive (Off) Active: Device's indicators are active showing their respective status ErrorStatus: Device's indicators are inactive unless an error occurs.
UserNameAvailable	User Name Available	Set if User Defined Name is available	
AccessPrivilegeAvailable	Access Privilege Available	Set if Heartbeat/Access Privilege is available	
MessageChannelSupported	Message Channel Supported	Set if the camera supports a Message channel	
TimestampSupported	Timestamp Supported	Set if the camera supports a timestamp register	
StringEncoding	String Encoding	String Encoding of the BRM	
FamilyRegisterAvailable	Family Register Available	Set if the camera supports the Family Name register	
SBRMSupported	SBRM Supported	Set if the camera supports a SBRM	
EndianessRegistersSupported	Endianess Registers Supported	Set if the camera supports the Protocol Endianess and Implementation Endianess registers	
WrittenLengthFieldSupported	Written Length Field Supported	Set to 1 if the camera sends the length_written field in the SCD section of the WriteMemAck command	

C.2 Analog Control

Name	Display Name	Description	Value
Gain	Gain (dB)	Gain applied to the image in dB	
GainAuto	Gain Auto	Controls the mode for automatic gain adjustment	Off Once Continuous
BlackLevel	Black Level (percent)	Analog black level (brightness) in percent	
BlackLevelEnabled	Black Level Enabled	Enables/disables black level adjustment	True False
	Black Level Auto	Controls the mode for automatic black level adjustment	Off Once Continuous
BalanceRatioSelector	Balance Ratio Selector	Selects which balance ratio to control (for White Balance)	Red Blue
BalanceRatio	Balance Ratio	Controls the ratio of the selected color component to a reference color component	
BalanceWhiteAuto	Balance White Auto	Controls the mode for automatic white balancing between color channels	Off Once Continuous
Gamma	Gamma	Controls the gamma correction of pixel intensity	
GammaEnabled	Gamma Enabled	Enables/disables gamma correction	True False
Sharpness	Sharpness	Sharpness of the image	
SharpnessEnabled	Sharpness Enabled	Enables/disables sharpness adjustment	True False
SharpnessAuto	Sharpness Auto	Controls the mode for automatic sharpness adjustment	Off Once Continuous
Hue	Hue (degrees)	Hue of the image in degrees	
HueEnabled	Hue Enabled	Enables/disables Hue	True False
	Hue Level Auto	Controls the mode for automatic hue adjustment	Off Once Continuous
Saturation	Saturation (percent)	Saturation of the image in percent	
	Saturation Enabled	Enables/disables saturation	True False
	Saturation Auto	Controls the mode for automatic saturation adjustment	Off Once Continuous

C.3 Image Format Control

Name	Display Name	Description	Value
SensorWidth	Sensor Width	Effective width of the sensor in pixels	
SensorHeight	Sensor Height	Effective height of the sensor in pixels	
MaxWidth	Max Width	Maximum width of the image in pixels	
MaxHeight	Max Height	Maximum height of the image in pixels	
Width	Width	Width of the image provided by the device in pixels	
Height	Height	Height of the image provided by the device in pixels	
OffsetX	Offset X	Vertical offset from the origin to the AOI in pixels	
OffsetY	Offset Y	Horizontal offset from the origin to the AOI in pixels	
ReverseX	Reverse X	Flip horizontally the image sent by the device. The AOI is applied after the flip	True False
	On Board Color Process Enabled	Enable FPGA processing (i.e., Gamma/Lookup table). This limits frame rate.	True False
PixelFormat	Pixel Format	Format of the pixel data (not all cameras support all formats)	Mono8, Mono12, Mono16, Raw8, Raw12, Raw16, RGB, YUV411, YUV422
PixelCoding	Pixel Coding	Coding of the pixels in the image	Mono Raw YUV RGB
PixelSize	Pixel Size	Size of a pixel in bits	8/12/16/24
PixelColorFilter	Pixel Color Filter	Type of color filter that is applied to the image	
TestImageSelector	Test Image Selector	Selects the type of test image that is sent by the camera	Off Test Image 1 Test Image 2
VideoMode	Video Mode	Current video mode	0 ... 8
PixelBigEndian	Pixel BigEndian	Set the pixel endianness for pixel format Mono16	True False
BinningHorizontal	Binning Horizontal	Number of horizontal pixels to combine together	
BinningVertical	Binning Vertical	Number of vertical pixels to combine together	
PixelDynamicRangeMin	Dynamic Range Min	Indicates the minimum pixel value transferred from the camera	
PixelDynamicRangeMax	Dynamic Range Max	Indicates the maximum pixel value transferred from the camera	

Name	Display Name	Description	Value
OnBoardColorProcessing	On Board Color Processing	Disabling On Board Color Processing bypasses the FPGA/color core, which disables image processing, but allows for faster frame rates.	True False

C.4 Acquisition Control

Name	Display Name	Description	Value
AcquisitionMode	Acquisition Mode	Sets the acquisition mode of the device	Continuous Single Frame Multi Frame
AcquisitionStart	Acquisition Start	Starts the acquisition of the device	Write Only
AcquisitionStop	Acquisition Stop	Stops the acquisition of the device at the end of the current frame	Write Only
AcquisitionFrameCount	Acquisition Frame Count	Number of frames to acquire in Multi Frame acquisition mode	
AcquisitionFrameRate	Acquisition Frame Rate (Hz)	Controls the acquisition rate (in Hertz) at which the frames are captured	
AcquisitionFrameRateControlEnabled	Acquisition Frame Rate Control Enabled	Enables manual control of the camera frame rate	True False
FrameRateAuto	Frame Rate Auto	Controls the mode for automatic frame rate adjustment	Off Continuous
TriggerSelector	Trigger Selector	Selects the type of trigger to configure. Derived from Exposure Mode.	Exposure Start/ Exposure Active
TriggerMode	Trigger Mode	Controls whether or not the selected trigger is active	Off On
TriggerSource	Trigger Source	Specifies the internal signal or physical input line to use as the trigger source. The selected trigger must have its Trigger Mode set to On.	Software Line x where x is a GPIO trigger pin
TriggerActivation	Trigger Activation	Specifies the activation mode of the trigger	Falling Edge Rising Edge
TriggerDelay	TriggerDelay (us)	Specifies the delay (in microseconds) to apply after the trigger reception before activating it	
TriggerDelayEnabled	Trigger Delay Enabled	Specifies whether or not the Trigger Delay is enabled	True False

Name	Display Name	Description	Value
ExposureMode	Exposure Mode (not all models support all modes)	Sets the operation mode of the exposure (shutter). Toggles the Trigger Selector. Timed = Exposure Start; Trigger Width = Exposure Active	Timed Trigger Width
ExposureTime	ExposureTime (us)	Exposure time in microseconds when Exposure Mode is Timed	
ExposureAuto	Exposure Auto	Sets the automatic exposure mode when Exposure mode is Timed	Off Once Continuous
pgrExposureCompensationAuto	Exposure Compensation Auto	Sets the automatic exposure compensation value mode	
pgrExposureCompensation	Exposure Compensation	The measured or target image plane illuminance in EV	
pgrAutoExposureCompensationLowerLimit	Auto Exposure Compensation Lower Limit	Lower limit of the auto exposure compensation value(EV) parameter	
pgrAutoExposureCompensationUpperLimit	Auto Exposure Compensation Upper Limit	Upper limit of the auto exposure compensation value(EV) parameter	

C.5 Digital Input Output Control

Name	Display Name	Description	Value
LineSelector	+ Line Selector	Selects the physical line (or GPIO pin) of the external device connector to configure.	Line 0 Line 1 Line 2 Line 3
LineMode	Line Mode	Controls whether the physical line is used to Input or Output a signal. Choices are dependent on which line is selected.	Input Trigger Strobe Output
LineSource	Line Source	Selects which input or output signal to output on the selected line. Line Mode must be Output.	Exposure Active External Trigger Active
LineInverter	Line Inverter	Controls the inversion of the signal of the selected input or output line	True False
StrobeEnabled	Strobe Enabled	Enables/disables strobe	True False
UserOutputValue	User Output Value	Sets the value of the user output selector	True = High False = Low
LineDebounceTime	Line Debounce Time	Sets the value of the selected line debouncer time in microseconds	
LineStatus	Line Status	Returns the current status of the selected input or output line	True = High False = Low
LineStatusAll	Line Status All	Returns the current status of all available line signals at time of polling in a single bitfield	

Name	Display Name	Description	Value
StrobeDuration	Strobe Duration	Outputs a specified number of pulses with programmable high and low duration	
ExposureActiveMode	Exposure Active Mode	If the camera supports rolling shutter with a global reset, the strobe can be set to active from the start of exposure of the first line to the end of exposure on the last line (Any pixel) or from the start of exposure on the last line to the end of exposure on the first line (All pixels).	00: Line 1 Exposed 01: Any pixel exposed (only for global reset mode) 10: All pixels exposed (same as Line 1 for global reset mode)
3.3VEnable	3.3 V Enable	Enable or disable 3.3 V output	

C.6 Transport Layer Control

Name	Display Name	Description
PayloadSize	Payload Size	Provides the number of bytes transferred for each image or chunk on the stream channel. This is mainly used by the application software to determine size of image buffers to allocate.
TransmitFailureCount	Transmit Failure Count	This contains a count of the number of failed frame transmissions that have occurred since the last reset. An error occurs if the camera cannot arbitrate for the bus to transmit image data and the image data FIFO overflows.
GenCPVersionMajor	GenCP Version Major	Major version of the specification.
GenCPVersionMinor	GenCP Version Minor	Minor version of the specification.
U3VVersionMajor	U3V Version Major	Major version of USB3 Vision.
U3VVersionMinor	U3V Version Minor	Minor version of USB3 Vision.
U3VCPSIRMAvailable	U3V SIRM Available	Set if the device supports at least one device streaming interface.
U3VCPeirMAvailable	U3V EIRM Available	Set if the device supports at least one device event interface.
U3VCPiIDC2Available	U3V IIDC2 Available	Set if the device supports IIDC2 register map.
U3VMaxCommandTransferLength	Max Command Transfer Length	Specifies the maximum supported command transfer length of the device.
U3VMaxAcknowledgeTransferLength	Max Ack Transfer Length	Specifies the maximum supported Acknowledge transfer length of the device.
U3VNumberOfStreamChannels	Number of Stream Channels	Number of stream channels and corresponding streaming interface register maps.
U3VCurrentSpeed	Current Speed	Specifies the current speed of the USB link.

C.7 User Set Control

Name	Display Name	Description	Value
CurrentUserSet	Current User Set	Indicates the user set that is currently in use. At initialization time, the camera loads the most recently saved user set	0 (default) 1 2
UserSetSelector	User Set Selector	Selects the user set to load or save	Default User Set 1 User Set 2
UserSetLoad	User Set Load	Loads the user set specified by the User Set Selector to the device and makes it active	Write Only
UserSetSave	User Set Save	Saves the user set specified by the User Set Selector to the non-volatile memory of the device	Write Only
DefaultUserSet	Default User Set	Selects the default user set as the default start up set	Default User Set 1 User Set 2

C.8 Chunk Data Control

Name	Display Name	Description	Value
ChunkModeActive	Chunk Mode Active	Activates the inclusion of Chunk data in the payload of the image.	
ChunkSelector	Chunk Selector	Selects which chunk data to enable or control	Image Image CRC Frame Counter Offset X Offset Y Width Height Exposure Time Gain Black Level
ChunkEnable	Chunk Enable	Enables the inclusion of the selected Chunk data in the payload of the image	

C.9 LUT Control

Name	Display Name	Description	Value
LUTSelector	LUT Selector	Selects which LUT to control. Typically not available when only a single LUT is supported.	Luminance Red Green Blue Device-Specific
LUTEnable	LUT Enable	Activates the selected LUT	True False
LUTIndex	LUT Index	Controls the index (offset) of the coefficient to access in the selected LUT	≥ 0
LUTValue	LUT Value	Returns the Value at entry LUTIndex of the LUT selected by LUTSelector	Device-specific
LUTValueAll	Lut Value All	Accesses all the LUT coefficients in a single access without using individual LUTIndex	Device-specific

D Control and Status Registers

Some features of the Blackfly USB 3.1 are accessible only using control and status registers (CSRs) that conform to the IICD 1.32 standard.

These include the following:

- Frame Buffer—[IMAGE_RETRANSMIT: 634h](#)
- Trigger Modes—[TRIGGER_MODE: 830h](#)
- Embedded Information—[FRAME_INFO: 12F8h](#)
- Initialize—[INITIALIZE: 000h](#)
- Video Mode Error—[VMODE_ERROR_STATUS: 628h](#)
- Camera Log—[CAMERA_LOG: 1D00 – 1DFFh](#)
- LED Control—[LED_CTRL: 1A14h](#)

A complete list of CSRs can be found in the [FLIR Machine Vision Camera Register Reference](#).

D.1 IMAGE_RETRANSMIT: 634h

This register provides an interface to the camera's frame buffer functionality.

Transmitting buffered data is available when continuous shot is disabled. Either One shot or Multi shot can be used to transmit buffered data when *Transfer_Data_Select* = 1. Multi shot is used for transmitting one or more (as specified by *Count_Number*) buffered images. One shot is used for retransmission of the last image from the retransmit buffer.

Image data is stored in a circular image buffer when *Image_Buffer_Ctrl* = 1. If the circular buffer overflows, the oldest image in the buffer is overwritten.

Transmitted data is always stored in the retransmit buffer. If a last or previous image does not exist, (for example, an image has not been acquired since a video format or mode change), the camera still transmits an image from the retransmit buffer, but its contents are undefined.

The image buffer is initialized when *Image_Buffer_Ctr* is written to '1'. Changing the video format, video mode, image_size, or color_coding causes the image buffer to be initialized and *Max_Num_Images* to be updated.

Format:

Field	Bit	Description
Image_Buffer_Ctrl	[0]	Image Buffer On/Off Control 0: OFF, 1: ON
Transfer_Data_Select	[1]	Transfer data path 0: Live data, 1: Buffered image data Ignored if ISO_EN=1
	[2-7]	Reserved
Max_Num_Images	[8-19]	Maximum number of images that can be stored in the current video format. Must be greater than zero. This field is read only.
Number_of_Images	[20-31]	The number of images currently in buffer. This field is read only.

D.2 DATA_FLASH_CTRL: 1240h

This register controls access to the camera's on-board flash memory. Each bit in the data flash is initially set to 1.

The user can transfer as much data as necessary to the offset address (1244h), then perform a single write to the control register to commit the data to flash. Any modified data is committed by writing to this register, or by accessing any other control register.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-5]	Reserved
Clean_Page	[6]	Read: 0: Page is dirty, 1: Page is clean Write: 0: No-op, 1: Write page to data flash
	[7]	Reserved
Page_Size	[8-19]	8 == 256 byte page 9 == 512 byte page
Num_Pages	[20-31]	11 == 2048 pages 13 == 8192 pages

D.3 DATA_FLASH_DATA: 1244h

This register provides the 32-bit offset to the start of where the data is stored in the flash memory.

Format:

Offset	Field	Bit	Description
1244h	DF_Data	[0-31]	32-bit offset to the start of data

D.4 GPIO_XTRA: 1104h

The GPIO_XTRA register controls when a strobe starts: relative to the start of integration (default) or relative to the time of an asynchronous trigger. If the camera supports a rolling shutter with a global reset, the strobe can be set to active from the start of exposure of the first line to the end of exposure on the last line (Any pixel) or from the start of exposure on the last line to the end of exposure on the first line (All pixels).

Format:

Field	Bit	Description
Strobe_Start	[0]	Current Mode 0: Strobe start is relative to start of integration (default in free running) 1: Strobe start is relative to external trigger (default in trigger mode)
	[1-12]	Reserved

Field	Bit	Description
Shutter_Mode	[13]	Read only 0: Rolling shutter mode 1: Global reset mode
Pixels_Exposed	[14-15]	00: Line 1 exposed 01: Any pixel exposed (only available for global reset mode) 10: All pixels exposed (same as Line 1 for global reset mode) 11: Invalid
	[16-31]	Reserved

D.5 OUTPUT_VOLTAGE_ENABLE: 19D0h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-30]	Reserved
ON_OFF	[31]	Enable or disable 3.3 V output on Blackfly GPIO pin 3. 0: Off, 1: On Default value is 0.

D.6 AE_ROI: 1A70 – 1A74h

AE_ROI is not supported on quad tap sensor cameras.

Note:

To calculate the base address for an offset CSR:

1. Query the offset inquiry register.
2. Multiple the value by 4. (The value is a 32-bit offset.)
3. Remove the 0xF prefix from the result. (i.e., F70000h becomes 70000h)

Format:

Offset	Name	Field	Bit	Description
1A70h	AE_ROI_CTRL	Presence_Inq	[0]	Presence of this feature 0:Not Available, 1: Available
			[1-5]	Reserved
		ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
			[7-31]	Reserved

Offset	Name	Field	Bit	Description
1A74h	AE_ROI_OFFSET		[0-31]	32-bit offset for the AE_ROI CSRs
Base + 0h	AE_ROI_UNIT_POSITION_INQ	Hposunit	[0-15]	Horizontal units for position
		Vposunit	[16-31]	Vertical units for position
Base + 4h	AE_ROI_UNIT_SIZE_INQ	Hunit	[0-15]	Horizontal units for size
		Vunit	[16-31]	Vertical units for size
Base + 8h	AE_ROI_POSITION	Left	[0-15]	Left position of ROI
		Top	[16-31]	Top position of ROI
Base + Ch	AE_ROI_SIZE	Width	[0-15]	Width of ROI
		Height	[16-31]	Height of ROI

D.7 FRAME_INFO: 12F8h

Field	Bit	Description	Frame-Specific Information
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available	
	[1-5]	Reserved	
ROI_Pos_Inq	[6]	Presence of image-specific information display 0: Not Available, 1: Available	
GPIO_State_Inq	[7]		
Strobe_Pat_Inq	[8]		
Frame_Count_Inq	[9]		
WB_CSR_Inq	[10]		
Exp_CSR_Inq	[11]		
Bright_CSR_Inq	[12]		
Shutter_CSR_Inq	[13]		
Gain_CSR_Inq	[14]		
Time_Inq	[15]		
CSR_Abs_Value	[16]	Toggles between displaying 32-bit relative or absolute CSR values. If absolute value not supported, relative value is displayed. 0: Relative, 1: Absolute This field is currently read-only	
	[17-21]	Reserved	

Field	Bit	Description	Frame-Specific Information
Insert_Info	[22]	Display image-specific information 0: Off 1: On	Region of Interest (ROI) position (See page 72)
	[23]		GPIO Pin State
	[24]		Strobe Pattern Counter
	[25]		Frame Counter
	[26]		White Balance CSR
	[27]		Exposure CSR
	[28]		Brightness CSR
	[29]		Shutter Value
	[30]		Gain CSR
	[31]		Timestamp (See page 71)

D.8 INITIALIZE: 000h

Format:

Offset	Name	Field	Bit	Description
000h	INITIALIZE	Initialize	[0]	If this bit is set to 1, the camera resets to its power up state and loads settings from the last save memory channel. This bit is self-cleared.
			[1-31]	Reserved

D.9 TEMPERATURE: 82Ch

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-19]	Reserved
Value	[20-31]	Value. In Kelvin (0°C = 273.15K) in increments of one-tenth (0.1) of a Kelvin

D.10 VMODE_ERROR_STATUS: 628h

Format:

Field	Bit	Description
Vmode_Error_Status	[0]	Error status of combination of video format, mode, frame rate and ISO_SPEED setting. 0: no error, 1: error This flag will be updated every time one of the above settings is changed by writing a new value.
	[1-31]	Reserved.

D.11 CAMERA_LOG: 1D00 – 1DFFh

Format:

Offset	Description
1D00..1DFF	Each byte is the hexadecimal representation of an ASCII character. The log is in reverse byte order, with the latest entry at the beginning of the log. The most significant byte of address 1D00h is the last byte in the log.

D.12 LED_CTRL: 1A14h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-22]	Reserved
LED_Ctrl	[23-31]	Enable or disable the LED 0x00: Off, 0x74: On

D.13 PIXEL_DEFECT_CTRL: 1A60h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-5]	Reserved
ON_OFF	[6]	Enable or disable FPGA pixel correction 0: Off, 1: On
	[7]	Reserved
Max_Pixels	[8-19]	Maximum number of pixels that can be corrected by the FPGA
Cur_Pixels	[20-31]	Current number of pixels that are being corrected by the FPGA