Chameleon

USB 2.0 Digital Camera

Technical Reference

Version 2.0

Revised 12/10/2013

USB 2.0

Point Grey Research® Inc.

12051 Riverside Way • Richmond, BC • Canada • V6W 1K7 •T (604) 242-9937 • www.ptgrey.com

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 Chameleon camera is 1 year. 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 Point Grey Research.

Trademarks

Point Grey Research, PGR, the Point Grey Research, Inc. logo, Blackfly, Bumblebee, Chameleon, Digiclops, Dragonfly, Dragonfly Express, Firefly, Flea, FlyCapture, Gazelle, Grasshopper, Ladybug, Triclops and Zebra are trademarks or registered trademarks of Point Grey Research, Inc. in Canada and other countries.

Table of Contents

Contacting Point Grey Research ..... **i**

1 Chameleon Specifications **1**

1.1 Chameleon Specifications 1

1.2 Description of the Data Flow 2

1.3 Handling Precautions and Camera Care 3

1.3.1 Case Temperature and Heat Dissipation 3

1.4 Analog-to-Digital Converter 5

2 Chameleon Installation **6**

2.1 Before You Install 6

2.1.1 Will your system configuration support the camera? 6

2.1.2 Do you have all the parts you need? 6

2.1.3 Do you have a downloads account? 6

2.2 Installing Your Interface Card and Software 7

2.3 Installing Your Camera 8

2.4 Configuring Camera Setup 9

2.4.1 Configuring Camera Drivers 9

2.4.2 Maximum Number of Cameras on a Single Bus 9

3 Tools to Control the Chameleon **10**

3.1 Using FlyCapture 10

3.1.1 FlyCap Program 10

3.1.2 Custom Applications Built with the FlyCapture API 10

3.2 Using Control and Status Registers 11

4 Chameleon Physical Interface **12**

4.1 Chameleon Physical Description 12

4.2 Chameleon Dimensions 13

4.3 Mounting with the Case or Mounting Bracket 14

4.3.1 Tripod Adapter Dimensions 14

4.4 Lens Mounting 15

4.4.1 Back Flange Distance 15

4.5 Dust Protection 16

4.6 Infrared Cut-Off Filters 17

4.7 Camera Interface and Connectors 18

4.7.1 USB 2.0 Connector 18

4.7.2 Interface Cables 18

4.7.3 Interface Card 18

4.7.4 General Purpose Input/Output (GPIO) 18

5 General Chameleon Operation 20
5.1 Powering the Camera 20
5.2 User Sets (Memory Channels) 20
5.3 Non-Volatile Flash Memory 21
5.4 Camera Firmware 21
5.4.1 Determining Firmware Version 21
5.4.2 Upgrading Camera Firmware 21
6 Input/Output Control 23
6.1 General Purpose Input/Output (GPIO) 23
6.2 GPIO Modes 24
6.2.1 GPIO Mode 0: Input 24
6.2.2 GPIO Mode 1: Output 24
6.2.3 GPIO Mode 2: Asynchronous (External) Trigger 24
6.2.4 GPIO Mode 3: Strobe 24
6.2.5 GPIO Mode 4: Pulse Width Modulation (PWM) 24
6.3 Programmable Strobe Output 25
6.4 Pulse Width Modulation (PWM) 25
6.5 Serial Communication 26
6.6 GPIO Electrical Characteristics 26
7 Image Acquisition 27
7.1 Asynchronous Triggering 27
7.1.1 Standard External Trigger (Mode 0) 28
7.1.2 Bulb Shutter Trigger (Mode 1) 29
7.1.3 Skip Frames Trigger (Mode 3) 30
7.1.4 Overlapped Exposure Readout Trigger (Mode 14) 31
7.2 External Trigger Timing 32
7.3 Camera Behavior Between Triggers 32
7.4 Changing Video Modes While Triggering 33
7.5 Asynchronous Software Triggering 34
7.6 Isochronous Data Transfer 34
8 Chameleon Attributes 35
8.1 Pixel Formats 35
8.1.1 Raw 35
8.1.2 Mono 35
8.1.3 RGB 35
8.1.4 YUV 35
8.1.5 Y16 (16-bit Mono) Image Acquisition 36
8.2 Video Modes Overview 37
8.2.1 Chameleon Video Modes 38

8.2.1.1 Standard Modes 38
8.2.1.2 Custom Modes 38
8.3 Frame Rates 40
8.3.1 Calculating Maximum Possible Frame Rate 40
8.4 Shutter Type 41
8.4.1 Global Shutter 41
8.5 Overview of Imaging Parameters 42
8.6 Brightness 43
8.7 Shutter Time 43
8.8 Gain 44
8.9 Auto Exposure 44
8.10 Sharpness 45
8.11 Gamma and Lookup Table 45
8.12 Embedded Image Information 47
8.13 White Balance 48
8.14 Bayer Color Processing 49
8.15 Hue 51
8.16 Saturation 51
9 Troubleshooting 52
9.1 Support 52
9.2 Camera Diagnostics 53
9.3 Status Indicator LED 54
9.4 Test Pattern 54
9.5 Blemish Pixel Artifacts 55
9.5.1 Pixel Defect Correction 55
9.6 Vertical Smear Artifact 56
9.6.1 Smear Reduction 56
9.7 Dark Current Accumulation 56
A FlyCapture API Code Samples 57
A.1 Setting a GPIO Pin to Strobe Using the FlyCapture API 57
A.2 Setting a Standard Video Mode, Format and Frame Rate Using the FlyCapture API 57
A.3 Asynchronous Hardware Triggering Using the FlyCapture API 57
A.4 Setting Brightness Using the FlyCapture API 58
A.5 Setting Shutter Using the FlyCapture API 58
A.6 Setting Gain Using the FlyCapture API 58
A.7 Setting Auto Exposure Using the FlyCapture API 59
A.8 Setting Sharpness Using the FlyCapture API 59
A.9 Setting Gamma Using the FlyCapture API 59
A.10 Setting White Balance Using the FlyCapture API 60

A.11 Accessing Raw Bayer Data using FlyCapture 60
A.12 Setting Hue Using the FlyCapture API 60
A.13 Setting Saturation Using the FlyCapture API 61
B FlyCapture SDK Examples 62
B.1 AsyncTriggerEx 62
B.2 BusEventsEx 62
B.3 CustomImageEx 62
B.4 ExtendedShutterEx 63
B.5 FlyCap2CameraControl 63
B.6 FlyCap2_GTKmm 63
B.7 FlyCap2MFC 64
B.8 FlyCapture2GUI 64
B.9 FlyCapture2SimpleGUI_WPF 64
B.10 FlyCapture2Test 64
B.11 GigEGrabEx 65
B.12 GrabCallbackEx 65
B.13 HighDynamicRangeEx 65
B.14 ImageEventEx 65
B.15 MultipleCameraEx 67
B.16 MultipleCameraWriteToDiskEx 67
B.17 MultiSyncEx 67
B.18 SaveImageToAviEx 67
B.19 SaveImageToFlashEx 67
B.20 SerialPortEx 68
C Isochronous Packet Format 69
C.1 Isochronous Packet Format 69
C.2 Isochronous Packet Format for Format 7 69
C.3 Isochronous Bandwidth Requirements: Format 0, Format 1, and Format 2 70
D Using Control and Status Registers 73
D.1 Modes 73
D.2 Values 73
D.3 Register Memory Map 73
D.4 Config ROM 75
D.4.1 Root Directory 75
D.4.2 Unit Directory 75
D.4.3 Unit Dependent Info 76
D.5 Calculating Base Register Addresses using 32-bit Offsets 77
D.6 Absolute Value Registers 78
D.6.1 Setting Absolute Value Register Values 78

D.6.2 Absolute Value Offset Addresses 78
D.6.3 Units of Value for Absolute Value CSR Registers 79
D.6.4 Determining Absolute Value Register Values 80
D.7 Inquiry Registers 81
D.7.1 Basic Functions Inquiry Registers 81
D.7.2 Feature Presence Inquiry Registers 81
D.7.3 Feature Elements Inquiry Registers 83
D.7.4 Video Format Inquiry Registers 85
D.7.5 Video Mode Inquiry Registers 85
D.7.6 Video Frame Rate Inquiry Registers 86
D.8 General Camera Operation 91
D.8.1 Memory Channel Registers 91
D.8.1.1 MEMORY_SAVE: 618h 92
D.8.1.2 MEM_SAVE_CH: 620h 92
D.8.1.3 CUR_MEM_CH: 624h 93
D.8.2 Device Information 93
D.8.2.1 SERIAL_NUMBER: 1F20h 93
D.8.2.2 MAIN_BOARD_INFO: 1F24h 93
D.8.2.3 VOLTAGE: 1A50h – 1A54h 94
D.8.2.4 CURRENT: 1A58h – 1A5Ch 94
D.8.2.5 TEMPERATURE: 82Ch 94
D.8.2.6 CAMERA_POWER: 610h 94
D.8.2.7 PIXEL_CLOCK_FREQ: 1AF0h 95
D.8.2.8 HORIZONTAL_LINE_FREQ: 1AF4h 95
D.8.3 Camera Memory 95
D.8.3.1 DATA_FLASH_CTRL: 1240h 95
D.8.3.2 DATA_FLASH_DATA: 1244h 95
D.8.3.3 IMAGE_RETRANSMIT: 634h 95
D.8.4 Firmware Information 96
D.8.4.1 FIRMWARE_VERSION: 1F60h 96
D.8.4.2 FIRMWARE_BUILD_DATE: 1F64h 97
D.8.4.3 FIRMWARE_DESCRIPTION: 1F68-1F7Ch 97
D.9 Input/Output Control 98
D.9.1 GPIO_CTRL_PIN: 1110h-1140h 98
D.9.2 GPIO_XTRA_PIN: 1114h-1144h 99
D.9.3 GPIO_CONTROL: 1100h 99
D.9.4 GPIO_STRPAT_CTRL: 110Ch 100
D.9.5 GPIO_STRPAT_MASK_PIN: 1118h-1148h 100
D.9.6 GPIO_XTRA: 1104h 100

D.9.7 Serial Input/Output Registers 101
D.10 Video Mode Control and Status Registers 105
D.10.1 FRAME_RATE: 83Ch 105
D.10.2 CURRENT_FRAME_RATE: 600h 105
D.10.3 CURRENT_VIDEO_MODE: 604h 106
D.10.4 CURRENT_VIDEO_FORMAT: 608h 106
D.10.5 FORMAT_7_RESIZE_INQ: 1AC8h 106
D.10.6 Inquiry Registers for Custom Video Mode Offset Addresses 107
D.10.6.1 Image Size and Position 107
D.10.6.2 COLOR_CODING_ID and COLOR_CODING_INQ 108
D.10.6.3 FRAME_INTERVAL_INQ 109
D.10.7 IMAGE_DATA_FORMAT: 1048h (IIDC 1.31) 109
D.11 Asynchronous Trigger Settings 111
D.11.1 TRIGGER_MODE: 830h 111
D.11.2 TRIGGER_DELAY: 834h 112
D.11.3 SOFTWARE_TRIGGER: 62Ch 112
D.12 Controlling Imaging Parameters 113
D.12.1 Imaging Parameters: 800h-888h 113
D.12.2 LUT: 1A40h – 1A44h (IIDC 1.31) 114
D.12.3 WHITE_BALANCE: 80Ch 115
D.12.4 BAYER_TILE_MAPPING: 1040h 116
D.12.5 SHUTTER: 81Ch 116
D.12.6 AUTO_EXPOSURE: 804h 117
D.12.6.1 AUTO_EXPOSURE_RANGE: 1088h 118
D.12.6.2 AUTO_SHUTTER_RANGE: 1098h 118
D.12.6.3 AUTO_GAIN_RANGE: 10A0h 119
D.12.6.4 AE_ROI: 1A70 – 1A74h 119
D.12.7 FRAME_INFO: 12F8h 120
D.13 Troubleshooting 121
D.13.1 Camera Diagnostics 121
D.13.1.1 INITIALIZE: 000h 121
D.13.1.2 TIME_FROM_INITIALIZE: 12E0h 121
D.13.1.3 XMIT_FAILURE: 12FCh 121
D.13.1.4 VMODE_ERROR_STATUS: 628h 121
D.13.1.5 LED_CTRL: 1A14h 122
D.13.1.6 TEST_PATTERN: 104Ch 122
D.13.2 PIXEL_DEFECT_CTRL: 1A60h 122
Revision History 123

Contacting Point Grey Research

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

Email	General questions about Point Grey Research Technical support (existing customers only)	
Knowledge Base	Find answers to commonly asked questions in our Knowledge Base	
Downloads	Download the latest documents and software	
Main Office	Point Grey Research, Inc. 12051 Riverside Way Richmond, BC, Canada V6W 1K7	**Tel:** +1 (604) 242-9937 **Toll Free** +1 (866) 765-0827 (North America only) **Fax:** +1 (604) 242-9938 **Email:** sales@ptgrey.com
USA		**Tel**: +1 (866) 765-0827 **Email:** na-sales@ptgrey.com
Europe and Israel	Point Grey Research GmbH Schwieberdinger Strasse 60 71636 Ludwigsburg Germany	**Tel:** +49 7141 488817-0 **Fax:** +49 7141 488817-99 **Email:** eu-sales@ptgrey.com
Distributors		
Japan	ViewPLUS Inc.	www.viewplus.co.jp
Korea	Cylod Co. Ltd.	www.cylod.com
China	LUSTER LightVision Tech. Co., Ltd.	www.lusterlighttech.com
Singapore, Malaysia & Thailand	Voltrium Systems Pte Ltd.	www.voltrium.com.sg
Taiwan	Apo Star Co., Ltd.	www.apostar.com.tw
United Kingdom	ClearView Imaging Ltd.	www.clearviewimaging.co.uk

About This Manual

This manual provides the user with a detailed specification of the Chameleon 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. (See Contacting Point Grey Research.)

This document is subject to change without notice.

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
Chameleon Specifications	General camera specifications and specific model specifications, and camera properties.
Chameleon Installation	Instructions for installing the Chameleon, as well as introduction to Chameleon configuration.
Tools to Control the Chameleon	Information on the tools available for controlling the Chameleon.
Chameleon Physical Interface	Information on the mechanical properties of the Chameleon.
General Chameleon Operation	Information on powering the Chameleon, 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.
Chameleon Attributes	Information on supported imaging parameters and their controls.
Troubleshooting	Information on how to get support, diagnostics for the Chameleon, 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: Using Control and Status Registers	Information on IIDC Control and Status Registers.

Document Conventions

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

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.

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.

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 Point Grey 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

1 Chameleon Specifications

1.1 Chameleon Specifications

Model	Version	MP	Imaging Sensor
CMLN-13S2C-CS **CMLN-13S2M-CS**	Color Mono	1.3 MP	▪ Sony ICX445 CCD, 1/3", 3.75 µm ▪ Global Shutter ▪ 18 FPS at 1296 x 964

	All Chameleon Models
Imaging Performance (EMVA 1288)	See the Imaging Performance Specification, which includes quantum efficiency, saturation capacity (full well depth), read noise, dynamic range and signal to noise ratio.
A/D Converter	12-bit
Video Data Output	8-bit and 16-bit digital data
Image Data Formats	Y8, Y16 (monochrome), 8-bit and 16-bit Raw Bayer data (color)
Partial Image Modes	Pixel binning and region of interest (ROI) modes
Image Processing	Color/Greyscale conversion, gamma, lookup table, white balance
Shutter	Global shutter; Automatic/manual/one-push/extended shutter modes 0.01 ms to >10 seconds (extended shutter mode)
Gain	Automatic/manual/one-push modes 0 dB to 24 dB
Gamma	0.50 to 4.00
White Balance	Automatic/manual/one-push modes
Digital Interface	5-pin Mini-B USB2.0 digital interface for camera control, video data transmission, and power
Transfer Rates	480 Mbit/s
GPIO	7-pin JST GPIO connector, 4 pins for trigger and strobe, 1 pin +3.3 V, 1 VEXT pin for external power
External Trigger Modes	IIDC Trigger Modes 0, 1, 3, and 14
Synchronization	Via external trigger or software trigger
Memory Channels	3 memory channels for custom camera settings
Flash Memory	256 KB non-volatile memory
Dimensions	25.5 mm x 44 mm x 41 mm (excluding lens holder and connectors)
Mass	37 grams (without optics or tripod mounting bracket)
Power Consumption	2 W, 4.745 to 5.25 V via Mini-B USB 2.0 interface or JST 7-pin GPIO connector
Machine Vision Standard	IIDC v1.31
Camera Control	Via FlyCapture SDK, CSRs, or third party software

	All Chameleon Models
Camera Updates	In-field firmware updates
Lens Mount	CS-mount (5 mm C-mount adapter included)
Temperature	Operating: 0° to 45°C; Storage: -30° to 60°C
Humidity	Operating: 20 to 80% (no condensation) ; Storage: 20 to 95% (no condensation)
Compliance	CE, FCC, RoHS
Operating System	Windows XP SP1
Warranty	1 year

1.2 Description of the Data Flow

The diagram below depicts the flow of image data on the Chameleon from capture, through manipulation, to output. The table that follows describes the steps in more detail.

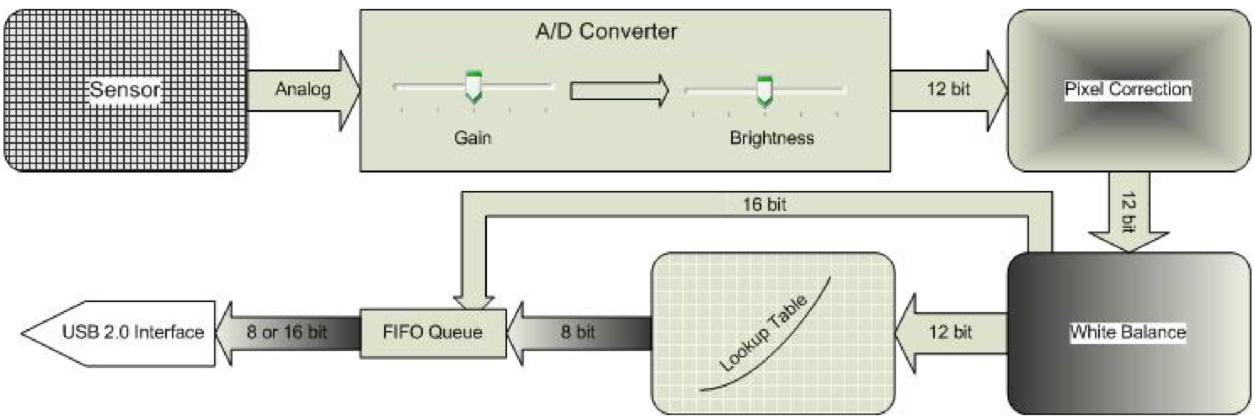


Figure 1.1: Data Flow Diagram for CMLN-13S2

Image Flow Step	Description
Sensor	Depending on the image size being captured, the Sony® ICX445 CCD sensor produces voltage signals in each pixel from the optical input.
Analog to Digital (A/D) Converter	The sensor's A/D Converter transforms pixel voltage into a 12-bit value, adjusting for gain and brightness in the process. The Chameleon supports automatic, manual and one-push gain control, and manual brightness control. Gain and brightness cannot be turned off.
Defect Correction	The camera firmware corrects any blemish pixels identified during manufacturing quality assurance by applying the average value of neighboring pixels. For more information, see Knowledge Base Article 314.
White Balance	In color models, color intensities can be adjusted manually to achieve more correct balance. White Balance is ON by default. If not ON, no white balance correction occurs.
Gamma/Lookup Table	Gamma correction can be applied manually, which results in adjustments to an 11-bit to 8-bit lookup table. By default, gamma adjustment is OFF, and no correction occurs. Gamma adjustment is not available if the camera is in a 16-bit image format.
FIFO Queue	The final output of image data is controlled in a first in, first-out (FIFO) queue.
USB 2.0 Interface	The camera's 5-pin Mini-B USB 2.0 port transfers data at a rate of 480 Mbit/s.

1.3 Handling Precautions and Camera Care

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

Your Point Grey digital 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.
- Users who have purchased a bare board camera should take the following additional protective measures:
  - Either handle bare handed or use non-chargeable gloves, clothes or material. Also, use conductive shoes.
  - Install a conductive mat on the floor or working table to prevent the generation of static electricity.
- When handling the camera unit, avoid touching the lenses. Fingerprints will 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
Solving problems with static electricity	Knowledge Base Article 42
Cleaning the imaging surface of your camera	Knowledge Base Article 66

1.3.1 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 1.1: Temperature Sensor Specifications

Accuracy	0.5°C
Range	-25°C to +85°C
Resolution	12-bits

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:

- CSRs—TEMPERATURE: 82Ch

1.4 Analog-to-Digital Converter

The camera sensor incorporates an analog to digital converter (ADC) to digitize the images produced by the CCD.

The Chameleon'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 with zeros.

A 12-bit conversion produces 4,096 possible digital image values between 0 and 65,520, left-aligned across a 2-byte data format. The four unused bits are padded with zeros.

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	0 dB to 18 dB
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. Image data is left-aligned across a 2-byte format. The least significant bits, which are the unused bits, are always zero.

For example, for a 12 bit output, the least significant 4 bits will be zeros in order to fill 2 bytes. E.g. 0xFFF0.

Model	ADC
CMLN-13S2M-CS	12-bit
CMLN-13S2C-CS	12-bit

2 Chameleon 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 XP SP1	2.0 GHz or equivalent	512 MB	AGP 128 MB	USB 2.0	Microsoft Visual Studio 2005 SP1 and SP1 Update for Vista (to compile and run example code using FlyCapture)

2.1.2 Do you have all the parts you need?

To install your camera you will need the following components:

- USB 2.0 cable (see Interface Cables)
- 7-pin GPIO cable (see General Purpose Input/Output (GPIO))
- CS-mount (or C-mount with adaptor) Lens (see Lens Mounting)
- Tripod adapter (optional) (see Mounting with the Case or Mounting Bracket)
- Interface card (see Interface Card)

Point Grey sells a number of the additional parts required for installation. To purchase, visit the Point Grey Webstore or the Products Accessories page.

2.1.3 Do you have a downloads account?

The Point Grey 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

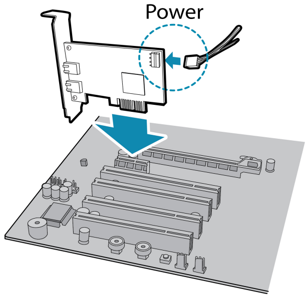
To access the downloads resources you must have a downloads account.

1. Go to the Point Grey downloads page.
2. Under **Register (New Users)**, complete the form, then click **Submit**.

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

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 the Point Grey 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

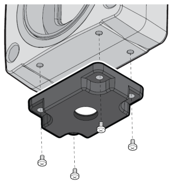
During the FlyCapture installation, you are prompted to select your interface driver.

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

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 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.

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 (optional)

GPIO can be used for power, trigger, pulse width modulation, serial input output, and strobe.

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: **Start-> FlyCapture 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 on the next page).

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

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 standard allows for 127 devices to be connected to a single bus. In practice, however, this limit may be further defined by the following considerations:

- Adequate power supply. The camera requires a nominal 5 volts (V) to operate effectively. While a standard, non-powered bus provides 500 milliamps (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 USB 2.0 bandwidth capacity is 480 Mbit/s. Depending on the operating configuration of the cameras and other devices, this bandwidth must be shared on the system.

Point Grey does not support the use of multiple USB 2.0 cameras streaming simultaneously on the same computer. There has been no rigorous qualification of the ability of various hardware platforms, operating systems, software, and drivers to handle multiple USB 2.0 image streams. Therefore, questions or troubleshooting of these issues cannot be addressed.

3 Tools to Control the Chameleon

The Chameleon's features can be accessed using various controls, including:

- FlyCapture SDK including API examples and the FlyCap program
- 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 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.

3.1.1 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 Point Grey 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.

3.1.2 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 Point Grey Imaging Products. 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

3.2 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.

For more information about camera registers, including the base register memory map, config ROM offsets and calculating register addresses, see Using Control and Status Registers.

A complete list of CSRs can be found in the *Point Grey Digital Camera Register Reference* available from the Downloads page.

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.

Not all features implement all modes.

Table 3.1: CSR Mode Control Descriptions

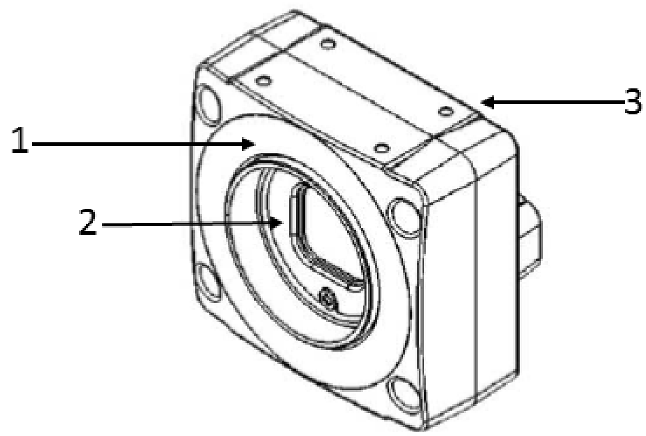
One_Push	ON_OFF	A_M_Mode	State
N/A	0	N/A	**Off state**. Feature will be fixed value state and uncontrollable.
N/A	1	1	**Auto control state**. Camera controls feature by itself continuously.
0	1	0	**Manual control state**. User can control feature by writing value to the value field.
1 (Self clear)	1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

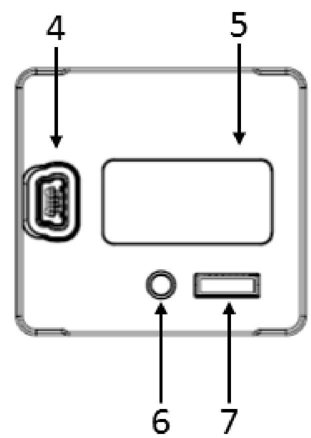
Values

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 Chameleon Physical Interface

4.1 Chameleon Physical Description





1. Lens holder (CS-mount)
Attach any CS-mount lens or other optical equipment. See Lens Mounting

2. Glass/IR filter system
See Dust Protection and Infrared Cut-Off Filters

3. M2x2 mounting holes
See Mounting with the Case or Mounting Bracket

4. USB 2.0 Mini-B vertical connector
The camera uses a USB 2.0 Mini-B vertical connector. See USB 2.0 Connector

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

6. Status LED
This light indicates the current state of the Chameleon operation. See Status Indicator LED

7. General Purpose I/O connector
The 7-pin GPIO connector is used for external triggering, strobe output or digital I/O. See Input/Output Control

4.2 Chameleon Dimensions

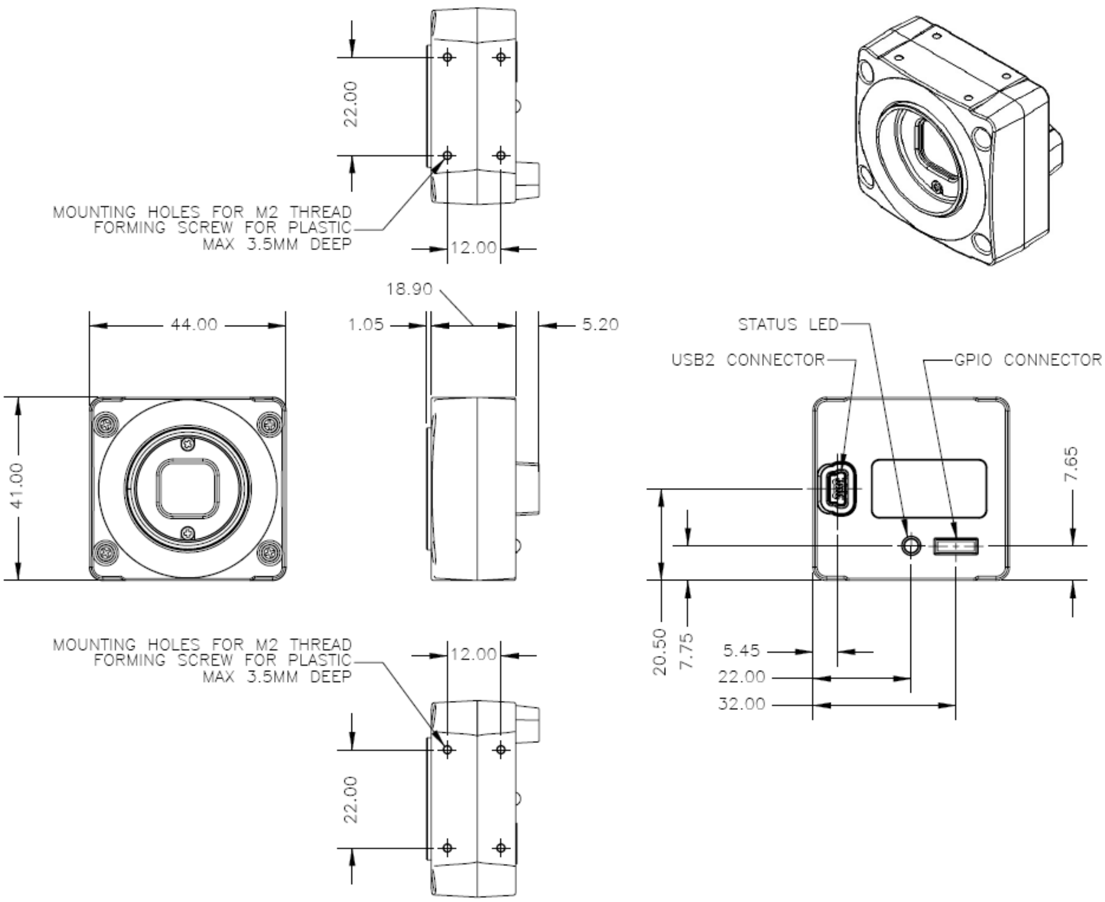


Figure 4.1: Chameleon Dimensional Diagram

To obtain 3D models, contact *support@ptgrey.com*.

4.3 Mounting with the Case or Mounting Bracket

Using the Case

The case is equipped with the following mounting holes:

- Four (4) M2x3.5 mm mounting holes on the top and 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 two (2) M3 and one (1) M2 mounting holes.

4.3.1 Tripod Adapter Dimensions

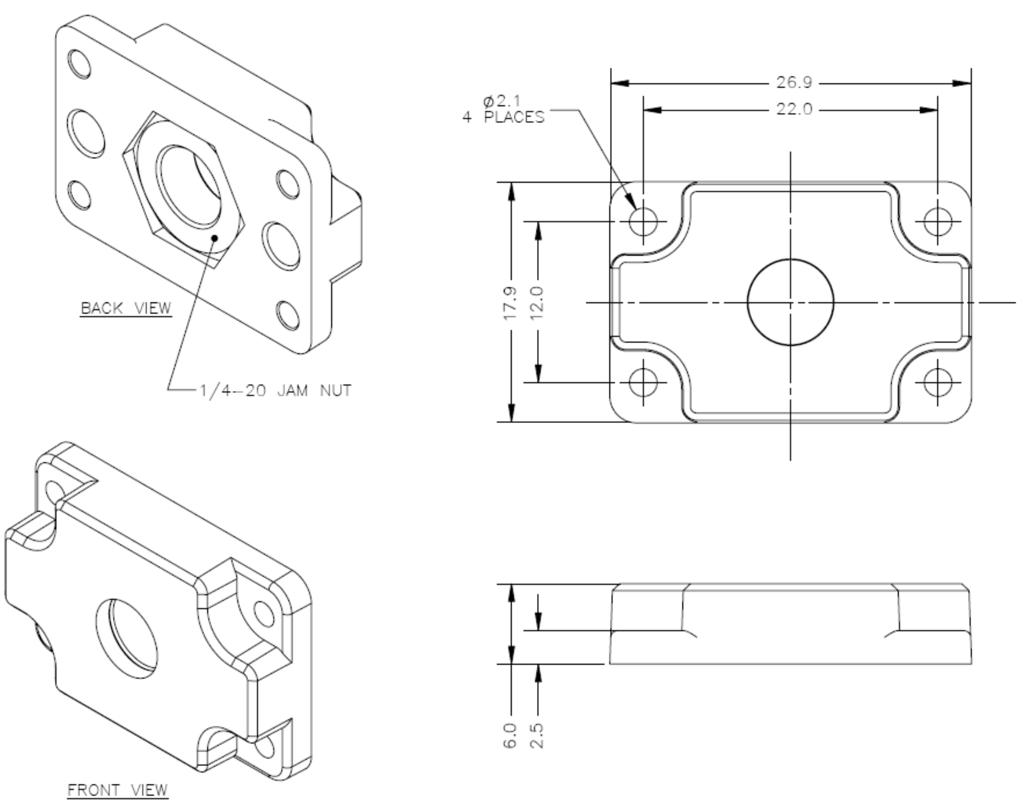


Figure 4.2: Tripod Adapter Dimensional Diagram

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 345

The lens mount is compatible with CS-mount lenses.

An M12 microlens holder can be obtained for use with board-level camera models.

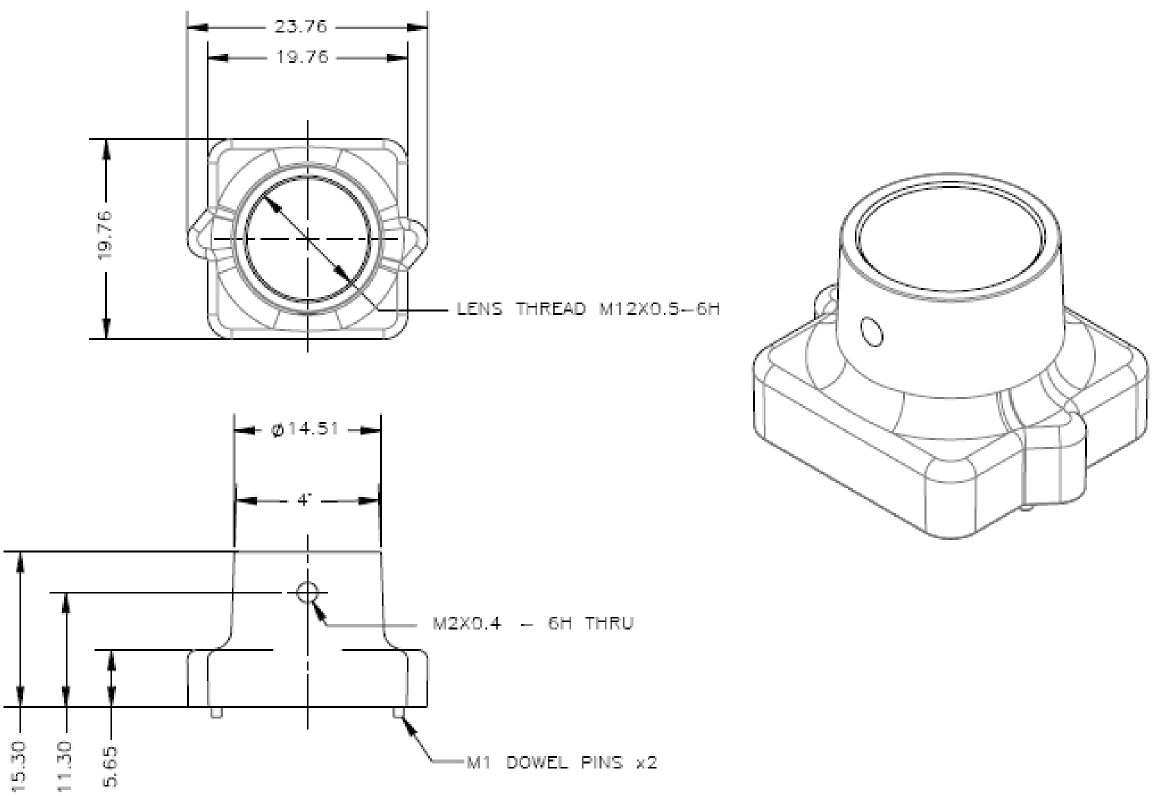


Figure 4.3: M12 Microlens Mount Dimensional Drawing (available separately for board-level models)

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 CS-mount BFD is 12.52 mm.

For more information about the IRC filter, see Infrared Cut-Off Filters.

4.5 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.

- *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 215
Selecting a lens for your camera	Knowledge Base Article 345

4.6 Infrared Cut-Off Filters

Point Grey 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.

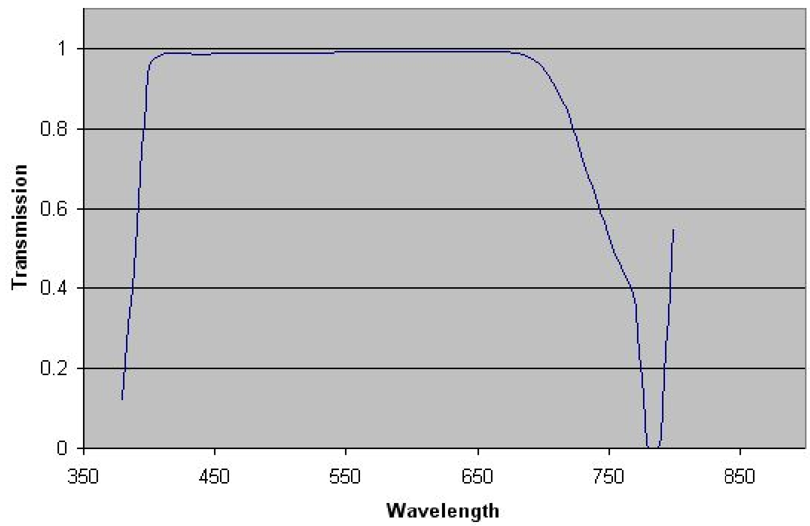


Figure 4.4: IR filter transmittance graph

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:

Type	Reflective
Material	Schott D 263 T
Physical Filter Size	14 mm x 14 mm
Glass Thickness	1.0 mm ±0.07 mm
Dimensional Tolerance	±0.1 mm

For more information, see Dust Protection.

Related Knowledge Base Articles

Title	Article
Removing the IR filter from a color camera	Knowledge Base Article 215

4.7 Camera Interface and Connectors

4.7.1 USB 2.0 Connector

The camera is equipped with a USB 2.0 Mini-B connector that is used for data transmission, camera control and power. For more detailed information, consult the USB 2.0 specification available from http://www.usb.org/developers/docs/.

The Chameleon is not backward compatible with a USB 1.1 interface. If the computer on which yo uwant to operate a Point Grey USB camera does not have a built-in USB 2.0 host controller, you can install a USB 2.0 PCI host adapter card. For more information refer to Knowledge Base Article 309.

4.7.2 Interface Cables

The maximum cable length between any USB node (for example, camera to USB, USB to hub, etc.) is 5.0 m, as indicated by the USB specification. Standard, shielded twisted pair copper cables must be used. For more information, refer to the USB.org website.

To purchase a recommended cable from Point Grey, visit the Point Grey Webstore or the Products Accessories page.

4.7.3 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).

The Chameleon is not backwards compatible with a USB 1.1 interface. If the computer on which you want to operate a Point Grey USB camera does not have a built-in USB 2.0 host controller, you can install a USB 2.0 PCI host adapter card.

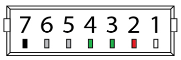
To purchase a compatible card from Point Grey, visit the Point Grey Webstore or the Products Accessories page.

Related Knowledge Base Articles

Title	Article
Using USB PCI 2.0 host adapter cards with USB cameras	Knowledge Base Article 309

4.7.4 General Purpose Input/Output (GPIO)

The camera is equipped with a 7-pin GPIO connector on the back of the case. The connector is made by JST (Mfg P/N: BM07B-SRSS-TB). The Development Kit contents include a pre-wired female connector; refer to the diagram below for wire color-coding.

Diagram	Color	Pin	Function	Description
	White	1	V_{EXT}	Allows the camera to be powered externally
	Red	2	+3.3V	Power external circuitry up to a total of 150 mA
	Green	3	IO0	Input/Output (Default Trigger_Scr)
	Green	4	IO1	Input/Output
	Grey	5	IO2	Input/Output/RS232 Transmit (TX)
	Grey	6	IO3	Input/Output/RS232 Receive (RX)
	Black	7	GND	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 Chameleon Operation

5.1 Powering the Camera

The power consumption specification is: 2 W, 4.745 to 5.25 V via Mini-B USB 2.0 interface or JST 7-pin GPIO connector.

The 5-pin USB 2.0 Mini-B vertical connector provides a power connection between the camera and the host computer. The ideal input voltage is 5V DC; however, the camera is designed to handle voltages between 4.75V and 5.25V DC.

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.

If Isochronous Data Transfer is enabled while the camera is powered down, the camera will automatically power itself up. However, disabling isochronous transmission does not automatically power-down the camera.

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
- 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, GPIO PWM modes
- Color Coding ID/Pixel Coding

To access user sets:

- CSRs—Memory Channel Registers

5.3 Non-Volatile Flash Memory

The camera has 256 KB non-volatile memory for users to store data.

To control flash memory:

- FlyCapture SDK example program—SaveImageToFlashEx
- CSRs—DATA_FLASH_CTRL: 1240h

Related Knowledge Base Articles

Title	Article
Storing data in on-camera flash memory	Knowledge Base Article 341

5.4 Camera Firmware

Firmware is programming that is inserted into the programmable read-only memory (programmable ROM) of most Point Grey 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
PGR software and firmware version numbering scheme/standards	Knowledge Base Article 96
Determining the firmware version used by a PGR camera	Knowledge Base Article 94
Should I upgrade my camera firmware or software?	Knowledge Base Article 225

5.4.1 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 Firmware Version register 1F60h

5.4.2 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 the Point Grey downloads site.

Before upgrading firmware:

- Install the SDK, downloadable from the Point Grey downloads site.
- Ensure that FlyCapture2.dll is installed in the same directory as UpdatorGUI3.
- Download the firmware file from the Point Grey downloads site.

To upgrade the firmware:

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

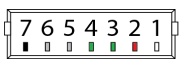
Do not disconnect the camera during the firmware update process.

6 Input/Output Control

6.1 General Purpose Input/Output (GPIO)

The camera is equipped with a 7-pin GPIO connector on the back of the case. The connector is made by JST (Mfg P/N: BM07B-SRSS-TB). The Development Kit contents include a pre-wired female connector; refer to the diagram below for wire color-coding.

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

Diagram	Color	Pin	Function	Description
	White	1	V_{EXT}	Allows the camera to be powered externally
	Red	2	+3.3V	Power external circuitry up to a total of 150 mA
	Green	3	IO0	Input/Output (Default Trigger_Scr)
	Green	4	IO1	Input/Output
	Grey	5	IO2	Input/Output/RS232 Transmit (TX)
	Grey	6	IO3	Input/Output/RS232 Receive (RX)
	Black	7	GND	Ground

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

For more information on camera power, see Powering the Camera.

For details on GPIO circuits, see GPIO Electrical Characteristics.

6.2 GPIO Modes

6.2.1 GPIO Mode 0: Input

When a GPIO pin is put into GPIO Mode 0 it is configured to accept external trigger signals. See Serial Communication.

6.2.2 GPIO Mode 1: Output

When a GPIO pin is put into GPIO Mode 1 it is configured to send output signals.

*Do **not** connect power to a pin configured as an output (effectively connecting two outputs to each other). Doing so can cause damage to camera electronics.*

6.2.3 GPIO Mode 2: Asynchronous (External) Trigger

When a GPIO pin is put into GPIO Mode 2, and an external trigger mode is enabled (which disables isochronous data transmission), the camera can be asynchronously triggered to grab an image by sending a voltage transition to the pin. See Asynchronous Triggering.

6.2.4 GPIO Mode 3: Strobe

A GPIO pin in GPIO Mode 3 outputs a voltage pulse of fixed delay, either relative to the start of integration (default) or relative to the time of an asynchronous trigger. A GPIO pin in this mode can be configured to output a variable strobe pattern. See Programmable Strobe Output.

6.2.5 GPIO Mode 4: Pulse Width Modulation (PWM)

When a GPIO pin is set to GPIO Mode 4, the pin outputs a specified number of pulses with programmable high and low duration. See Pulse Width Modulation (PWM).

6.3 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 200
GPIO strobe signal continues after isochronous image transfer stops	Knowledge Base Article 212
Setting a GPIO pin to output a strobe signal pulse pattern	Knowledge Base Article 207

6.4 Pulse Width Modulation (PWM)

When a GPIO pin is set to PWM (GPIO Mode 4), the pin will output a specified number of pulses with programmable high and low duration.

The pulse is independent of integration or external trigger. There is only one real PWM signal source (i.e. two or more pins cannot simultaneously output different PWMs), but the pulse can appear on any of the GPIO pins.

The units of time are generally standardized to be in ticks of a 1.024 MHz clock. A separate GPIO pin may be designated as an "enable pin"; the PWM pulses continue only as long as the enable pin is held in a certain state (high or low).

The pin configured to output a PWM signal (PWM pin) remains in the same state at the time the 'enable pin' is disabled. For example, if the PWM is in a high signal state when the 'enable pin' is disabled, the PWM pin remains in a high state. To re-set the pin signal, you must re-configure the PWM pin from GPIO Mode 4 to GPIO Mode 1.

To control PWM:

- CSRs—GPIO_CTRL_PIN: 1110h-1140h and GPIO_XTRA_PIN: 1114h-1144h

6.5 Serial Communication

The camera is capable of serial communications at baud rates up to 115.2 Kbps via the on-board serial port built into the camera's GPIO connector. The serial port uses TTL digital logic levels. If RS signal levels are required, a level converter must be used to convert the TTL digital logic levels to RS voltage levels.

B&B Electronics part number 232LPTTL can be used for this conversion.

Related Knowledge Base Articles

Title	Article
Configuring and testing the RS-232 serial port	Knowledge Base Article 151

SIO Buffers

The transmit and receive buffers have a 64 byte maximum.

6.6 GPIO Electrical Characteristics

The Chameleon GPIO pins are bi-directional. When configured as outputs, they operate as open collector transistor logic. As inputs, the lines are internally pulled up to 3.3 V.

When configured as inputs, the pins are internally pulled high using weak pull-up resistors to allow easy triggering of the camera by simply shorting the pin to ground (GND). Inputs can also be directly driven from a 3.3 V or 5 V logic output. The inputs are protected from both over and under voltage. It is recommended, however, that they only be connected to 5 V or 3.3 V digital logic signals. When configured as outputs, each line can sink 10 mA of current. To drive external devices that require more, consult Knowledge Base Article 200 for information on buffering an output signal using an optocoupler:

The V_{EXT} pin (Pin 1) allows the camera to be powered externally. The voltage limit is 4.75-5.25 V.

The +3.3 V pin (Pin 2) is fused at 200 mA. External devices connected to Pin 1 should not attempt to pull anything greater than that.

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 camera register write (software trigger).

Chameleon Supported Trigger Modes	
Model	**Mode**
All	Standard External Trigger (Mode 0)
All	Bulb Shutter Trigger (Mode 1)
All	Skip Frames Trigger (Mode 3)
All	Overlapped Exposure Readout Trigger (Mode 14)

To access trigger modes:

- FlyCapture API—AsyncTriggerEx
- CSRs—TRIGGER_MODE: 830h

7.1.1 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 Shutter value 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, this is possible using Overlapped Exposure Readout Trigger (Mode 14).

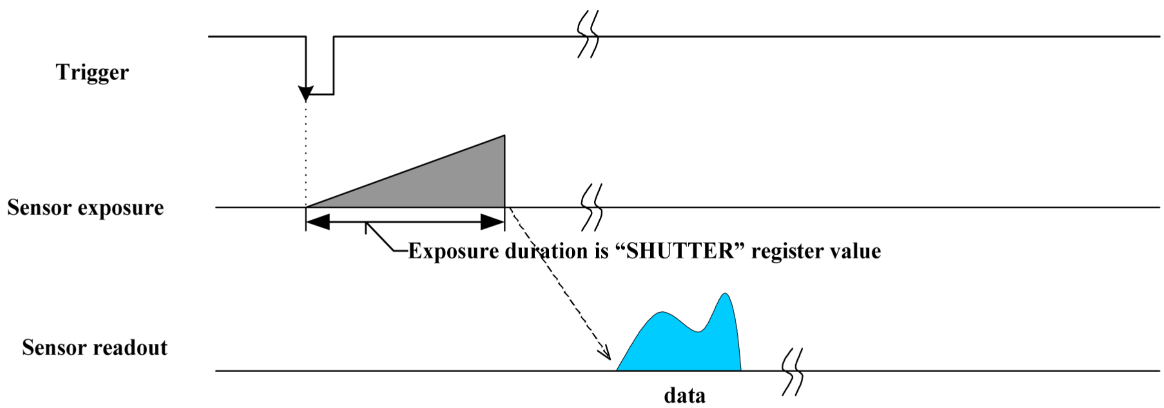


Figure 7.1: Trigger Mode 0 ("Standard External Trigger Mode")

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_0
Parameter	[20-31]	None

7.1.2 Bulb Shutter Trigger (Mode 1)

Also known as 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.

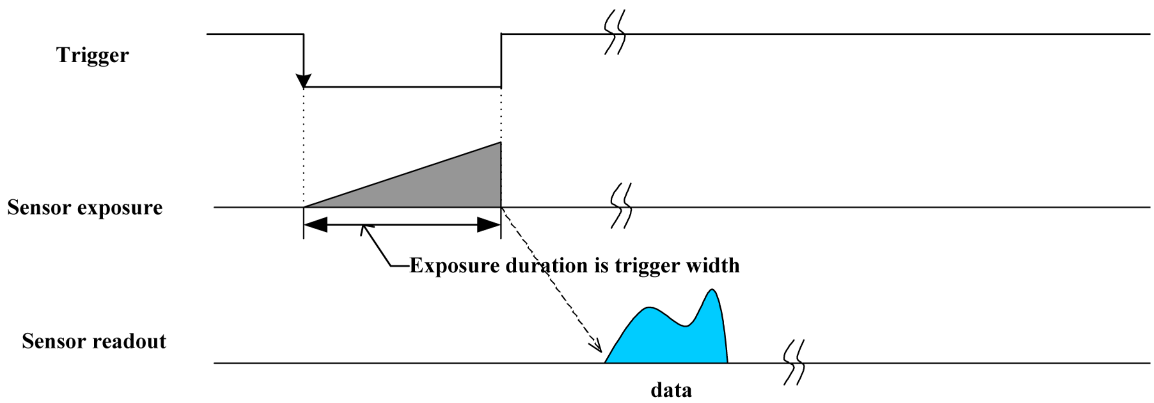


Figure 7.2: Trigger Mode 1 (“Bulb Shutter Mode”)

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

7.1.3 Skip Frames Trigger (Mode 3)

Trigger Mode 3 allows the user to put the camera into a mode where the camera only transmits one out of N specified images. This is an internal trigger mode that requires no external interaction. Where N is the parameter set in the Trigger Mode, the camera will issue a trigger internally at a cycle time that is N times greater than the current frame rate. As with Trigger Mode 0, the Shutter value describes integration time.

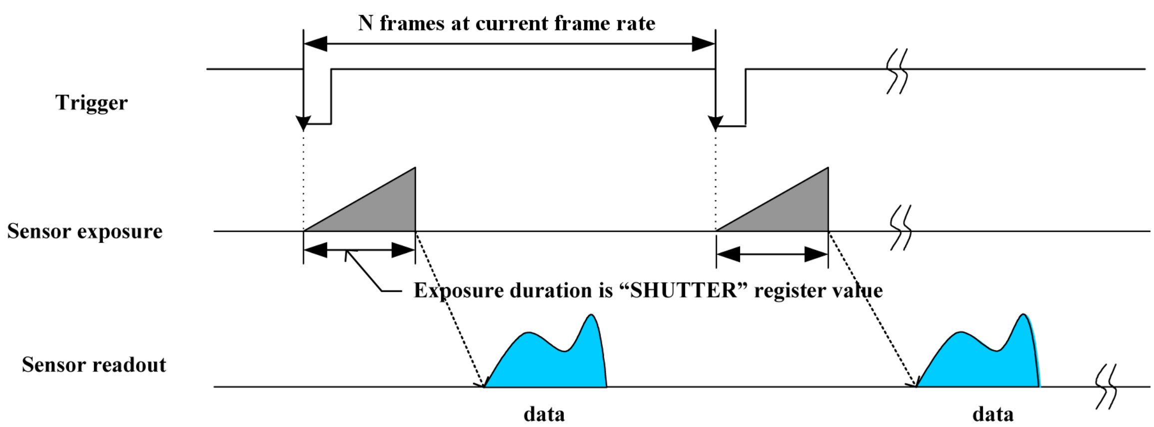


Figure 7.3: Trigger Mode 3 ("Skip Frames Mode")

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_3
Parameter	[20-31]	*N* 1 out of N images is transmitted. Cycle time N times greater than current frame rate

7.1.4 Overlapped Exposure Readout Trigger (Mode 14)

Trigger Mode 14 is a vendor-unique trigger mode that is very similar to Trigger Mode 0, 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 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 will start 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 will be 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.

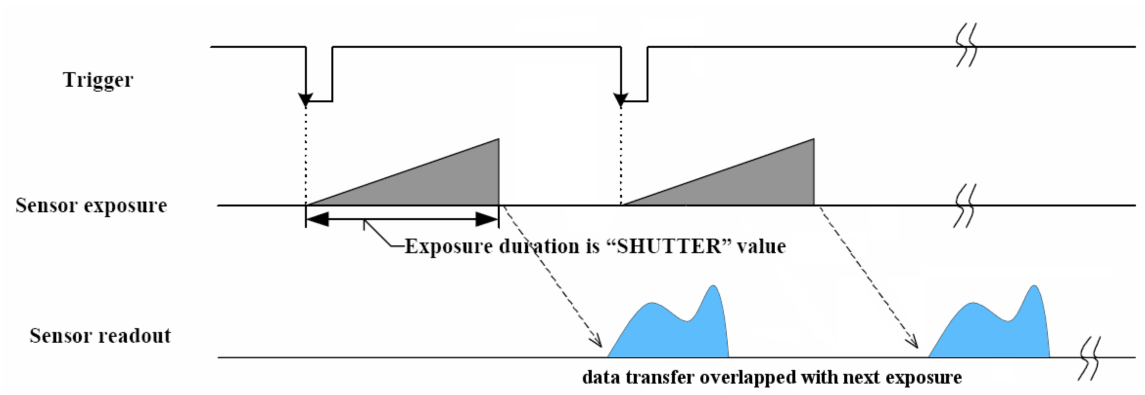
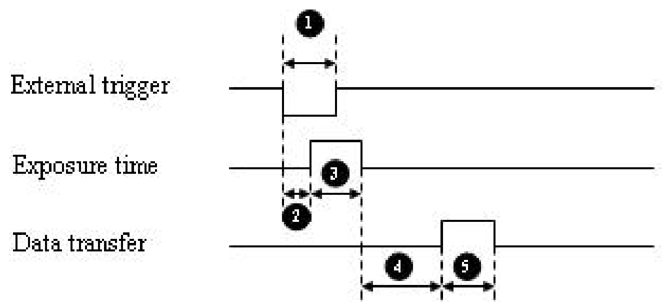


Figure 7.4: Trigger Mode 14 (“Overlapped Exposure/Readout Mode”)

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

7.2 External Trigger Timing

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



1. Trigger Pulse
2. Propagation Delay
3. Exposure Time
4. Sensor Readout
5. Data Transfer

Figure 7.5: External trigger timing characteristics

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 oscilliscope 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.

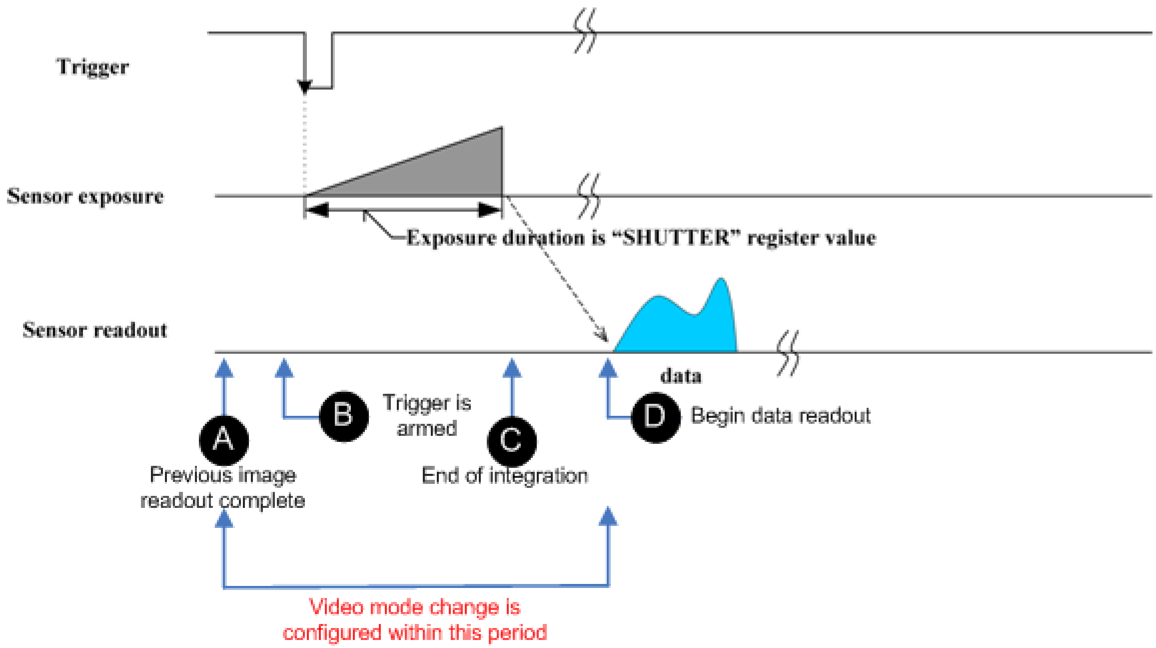


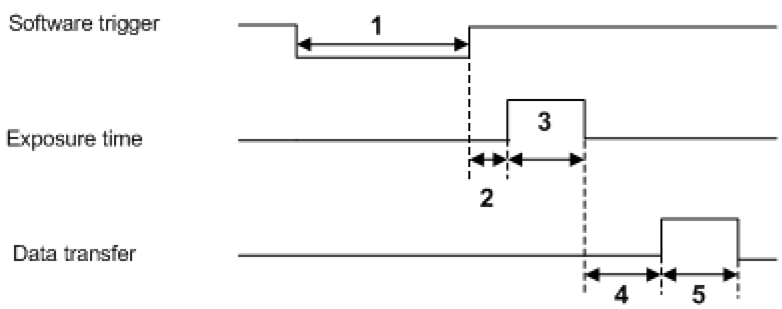
Figure 7.6: Relationship Between External Triggering and Video Mode Change Request

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 will attempt 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.

7.5 Asynchronous Software Triggering

Shutter integration can be initiated by a software trigger via SOFTWARE_TRIGGER: 62Ch.

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



1. Software Trigger
2. Trigger Latency
3. Exposure Time
4. Sensor Readout
5. Data Transfer

Figure 7.7: 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.

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.

7.6 Isochronous Data Transfer

Isochronous transmission is the transfer of image data from the camera to the PC in a continual stream that is regulated by an internal clock. Isochronous transfers on the bus guarantee timely delivery of data, but not necessarily integrity of data.

For more information about isochronous transmission, including packet formats and bandwidth requirements, see Isochronous Packet Format.

8 Chameleon 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 Chameleon's Analog-to-Digital Converter, which digitizes the images, is configured to a fixed bit output (12-bit). 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 with zeros.

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 disables image processing, such as gamma/LUT and color encoding, but allows for faster frame rates.

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. The 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. Point Grey cameras use packed YUV.

Related Knowledge Base Articles

Title	Article
Understanding YUV data formats	Knowledge Base Article 313

8.1.5 Y16 (16-bit Mono) Image Acquisition

The camera can output Y16 (16 bits-per-pixel) mono images. Because the camera's A/D converter is less than 16 bits, unused bits are zero.

Related Knowledge Base Articles

Title	Article
Method for converting signal-to-noise ratio (SNR) to/from bits of data	Knowledge Base Article 170

The data format for Y16 images is controlled by the Y16_Data_Format field of the IMAGE_DATA_FORMAT: 1048h (IIDC 1.31).

The PGM file format can be used to correctly save 16-bit images. Although the availability of photo manipulation/display applications that can correctly display true 16-bit images is limited, XV in Linux and Adobe Photoshop are two possibilities.

8.2 Video Modes Overview

The camera implements a number of Format 7 customizable video modes. These modes, which may increase frame rate and image intensity, operate by selecting a specific region of interest (ROI) of the image, or by configuring the camera to aggregate pixel values using a process known as “binning.” Some modes implement a combination of ROI and binning.

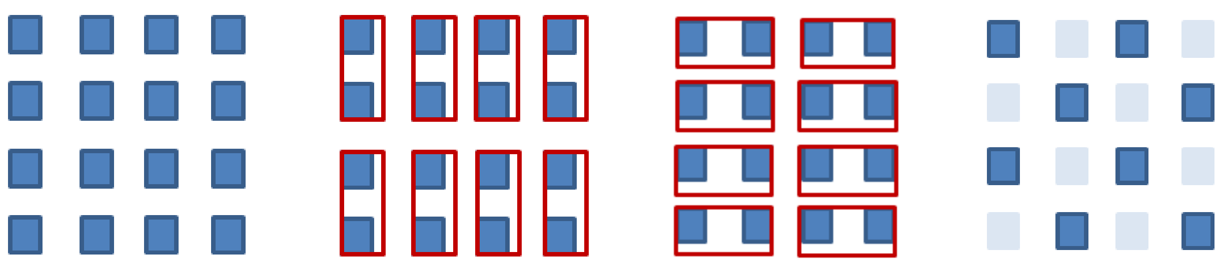
On Point Grey cameras, binning refers to the aggregation of pixels. Analog binning is aggregation that occurs before the analog to digital conversion. Digital binning is aggregation that occurs after the analog to digital conversion. Unless specified otherwise, color data is maintained in binning modes.

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 half.

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. 2x subsampling skips every second pixel horizontally and vertically.



Full Pixels | 2x Vertical Binning | 2x Horizontal Binning | 2x Subsampling

Figure 8.1: Aggregation and Decimation of Pixels

Changing the size of the image or the pixel encoding format requires the camera to be stopped and restarted. Ignoring the time required to do this in software (tearing down, then reallocating, image buffers, write times to the camera, etc.), the maximum amount of time required for the stop/start procedure is slightly more than one frame time.

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).

Additional binning information can be obtained by reading FORMAT_7_RESIZE_INQ: 1AC8h. The implementation of Format 7 modes and the frame rates that are possible are not specified by the IIDC, and are subject to change across firmware versions.

Pixel correction is not done in any of the binning modes.

8.2.1 Chameleon Video Modes

8.2.1.1 Standard Modes

This section lists the different video formats, modes and frame rates that are supported by the Chameleon. Refer to the Customizable Formats and Modes for a list of supported partial image (Format 7) modes. These standard modes are controlled using the following IIDC registers:

- CURRENT_VIDEO_FORMAT: 608h
- CURRENT_VIDEO_MODE: 604h
- CURRENT_FRAME_RATE: 600h

Modes	1.875 FPS	3.75 FPS	7.5 FPS	15 FPS	30 FPS*
640x480 Y8	•	•	•	•	•
640x480 Y16	•	•	•	•	•
1280x960 Y8	•	•	•	•	
1280x960 Y16	•	•	•		
*Monochrome output only. Color data is removed due to pixel binning.					

8.2.1.2 Custom Modes

The table below outlines the Format 7 custom image modes that are supported by the Chameleon. The implementation of these modes and the frame rates that are possible are not specified by the IIDC, and are subject to change across firmware versions.

Mode 0, Mode 1 and Mode 2 are region of interest (sub-window) modes that allow the user to only transmit a selected area of the image.

Mode 1 and Mode 2 are also pixel binning modes. Mode 1 implements 2X vertical and 2X horizontal binning, which lowers the image resolution by a factor of 4. Mode 2 implements 2X vertical binning only, which lowers resolution by a factor of 2. The binning in Mode 1 and Mode 2 is average binning and therefore pixel intensity does not increase in these modes.

The sizes and frame rates supported by monochrome (BW) models are identical to those of the color model specified below, with the exception that only Mono8 and Mono16 are supported.

Mode	Pixel Format	Unit Size (H,V)	Min BPP	Max BPP	1280 x 960	640 x 480	320 x 240	160 x 120
0	Raw8	8,2	244	2928	18	24	29	31
0	Raw16	8,2	488	3904	12	24	29	32
1	Mono8	4,2	124	1240	-	33	33	32
1	Mono16	4,2	244	2684	-	33	33	33
2	Mono8	8,2	244	2684	-	33	33	32
2	Mono16	8,2	488	3904	-	33	33	33

When outputting in Raw8 or Raw16 format, the camera outputs color data only in 1280 x 960 resolution. In lower resolutions, the camera performs pixel binning, which destroys the Bayer tile pattern.

8.3 Frame Rates

The current base frame rate is controlled using the CURRENT_FRAME_RATE: 600h. The Chameleon allows users to further “fine-tune” the frame rates of their cameras using the FRAME_RATE: 83Ch. This is particularly useful for capturing an image stream at a different frame rate than those outlined in the Supported Data Formats and Modes section, and can be useful for synchronizing to 50 Hz light sources, which can cause image intensity fluctuations due to the light source oscillations being out of sync with the frame rate.

For example, users may wish to play an image stream back on a PAL-based system that displays at 25 FPS. To do this, set the CURRENT_FRAME_RATE to 30 FPS, set the A_M_Mode bit [7] of the FRAME_RATE register 0x83C to zero (manual), then adjust the value using the Value field or using the ABS_VAL_FRAME_RATE register (recommended).

8.3.1 Calculating Maximum Possible Frame Rate

Theoretically, the maximum achievable frame rate for each camera on the network depends on available bandwidth, bytes per pixel, and resolution.

Bytes per pixel (BPP) is related to pixel format.

- 8-bit = 1 BPP
- 12-bit = 1.5 BPP
- 16-bit = 2 BPP
- 24-bit = 3 BPP

The theoretical frame rate (FPS) that can be achieved can be calculated as follows:

Frame Rate in FPS = (Bandwidth / (W x H x BPP)) / Number of Cameras

An example for CMLN-13S2:

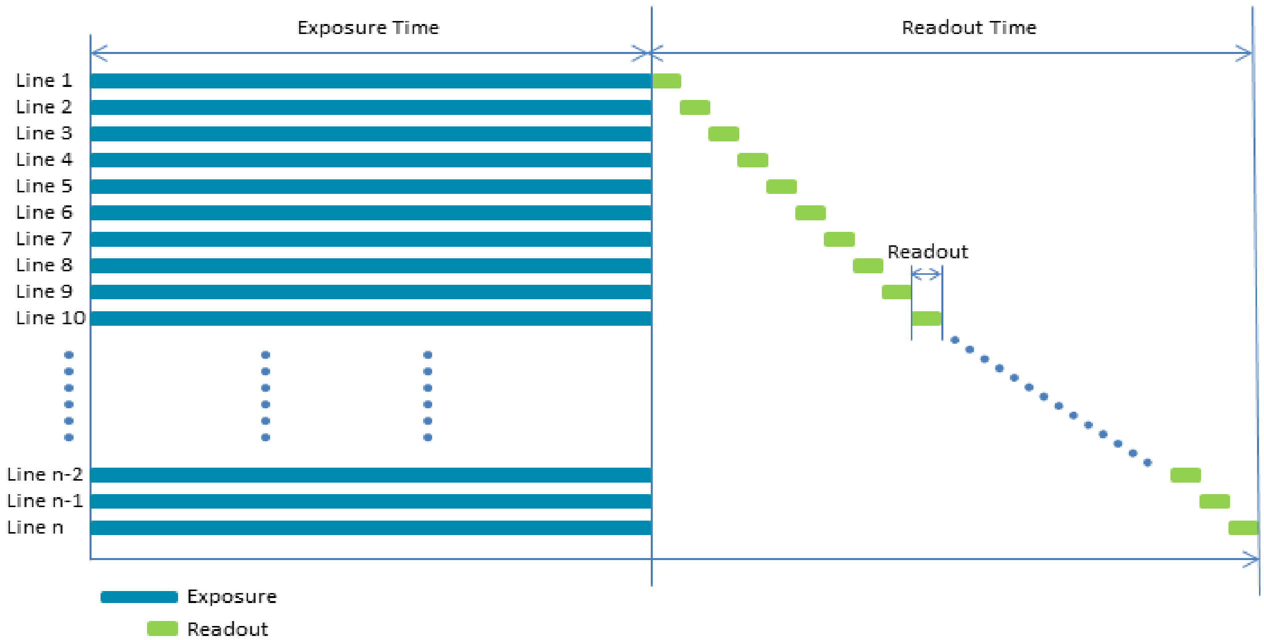
Assuming a 1032 x 776 image, with a Mono16 pixel format, using 31 MB/s bandwidth, the calculation would be:

Frame Rate = (Bandwidth / (W x H x BPP)) / Number of Cameras
Frame Rate = (31040248 / (1032 x 776 x 2)) / 1
Frame Rate = 19.38 FPS

8.4 Shutter Type

8.4.1 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.



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

8.5 Overview of Imaging Parameters

The camera supports control over the following imaging parameters:

Imaging Parameter	Register Control	FlyCapture API Sample Code
Brightness	Imaging Parameters: 800h-888h	Setting Brightness Using the FlyCapture API
Shutter Time	SHUTTER: 81Ch	Setting Shutter Using the FlyCapture API
Gain	Imaging Parameters: 800h-888h	Setting Gain Using the FlyCapture API
Auto Exposure	AUTO_EXPOSURE: 804h	Setting Auto Exposure Using the FlyCapture API
Sharpness	Imaging Parameters: 800h-888h	Setting Sharpness Using the FlyCapture API
Gamma and Lookup Table	Imaging Parameters: 800h-888h	Setting Gamma Using the FlyCapture API
	LUT: 1A40h – 1A44h (IIDC 1.31)	
Embedded Image Information	FRAME_INFO: 12F8h	
White Balance (color models only)	WHITE_BALANCE: 80Ch	Setting White Balance Using the FlyCapture API
Bayer Color Processing (color models only)	BAYER_TILE_MAPPING: 1040h	Accessing Raw Bayer Data using FlyCapture
Hue (color models only)	Imaging Parameters: 800h-888h	Setting Hue Using the FlyCapture API
Saturation (color models only)	Imaging Parameters: 800h-888h	Setting Saturation Using the FlyCapture API

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

There are three modes:

CSR Control

Mode	Description
On/Off	Determines if the feature is on. If off, values are fixed and not controllable.
Auto/Manual	If the feature is on, determines if the feature is in automatic or manual mode. If manual, values can be set.
One Push	If the feature is in manual mode, the camera executes once automatically and then returns to manual mode.

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

8.6 Brightness

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

The camera supports brightness control.

To adjust brightness:

- FlyCapture API—Setting Brightness Using the FlyCapture API
- CSRs—Imaging Parameters: 800h-888h

8.7 Shutter Time

The Chameleon supports Automatic, Manual, and One Push control of the image sensor shutter time.

Shutter 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 shutter time to double (e.g. 66 ms to 133 ms).

The maximum shutter 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 shutter time increase.

The maximum shutter time may only be available when operating the camera in Format 7 Mode 7. For more information, see Video Modes Overview.

The terms "integration", "exposure" and "shutter" are interchangeable.

The time between the end of shutter for consecutive frames is always constant. However, if the shutter time is continually changing (e.g. being controlled by Auto Exposure), the time between the beginning of consecutive integrations will change. If the shutter 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 shutter value take effect with the next captured image, or the one after next. Changes to shutter in asynchronous trigger mode generally take effect on the next trigger.

To adjust shutter:

- FlyCapture API—Setting Shutter Using the FlyCapture API
- CSRs—Imaging Parameters: 800h-888h

To enable extended shutter:

- FlyCapture SDK example program—ExtendedShutterEx

8.8 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 Chameleon supports Automatic and One Push 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.

Increasing gain also increases image noise, which can affect image quality. To increase image intensity, try adjusting the lens aperture (iris) and Shutter Time time first.

To adjust gain:

- FlyCapture API—Setting Gain Using the FlyCapture API
- CSRs—Imaging Parameters: 800h-888h

8.9 Auto Exposure

Auto exposure allows the camera to automatically control shutter 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 Shutter and Gain. This mode is achieved by setting Auto Exposure to Off, or by setting Shutter and Gain to Manual.
Manual Exposure Control	The camera automatically modifies Shutter and Gain to try to match the average image intensity to the Auto Exposure value. This mode is achieved by setting Auto Exposure to Manual and either/both of Shutter and Gain to Automatic.
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, Shutter, and Gain to Automatic. In this mode, the value reflects the average image intensity.

Auto Exposure can only control the exposure when Shutter and/or Gain are set to Automatic. 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 shutter-before-gain heuristic to try and maximize the signal-to-noise ratio by favoring a longer shutter time over a larger gain value.

In absolute mode, an exposure value (EV) of 0 indicates the average intensity of the image is 18% of 1023 (18% gray).

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 shutter values to be used by the automatic exposure controller which is generally some subset of the entire shutter 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.

To control auto exposure:

- FlyCapture API—Setting Auto Exposure Using the FlyCapture API
- CSRs—AUTO_EXPOSURE: 804h and AE_ROI: 1A70 – 1A74h

8.10 Sharpness

The Chameleon 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 1000, the pixel is sharpened; for values less than 1000 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:

- FlyCapture API—Setting Sharpness Using the FlyCapture API
- CSRs—Imaging Parameters: 800h-888h

8.11 Gamma and Lookup Table

The camera supports gamma and lookup table (LUT) functionality.

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. 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:

```
OUT = 255*(IN/255)^1/gamma
```

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

Alternatively, the camera has a 9-bit input lookup table that produces a 9-bit output. The LUT has two banks that the user can select between. In RGB and YUV pixel formats, the LUT has three channels for red, green, and blue. In monochrome and raw formats, there is a single channel, regardless of color or monochrome sensor. The LUT is available only in 8 bit/pixel formats.

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^{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^{Input_Depth}-1)
C(1)E(0) … … C(1)E(2^{Input_Depth}-1)

... ...
C(Number_of_Channels-1)E(0) ... ... C(Number_of_Channels-1) $E(2^{Input\_Depth}-1)$

Related Knowledge Base Articles

Title	Article
How is gamma calculated and applied?	Knowledge Base Article 391

To adjust gamma:

- FlyCapture API—Setting Gamma Using the FlyCapture API
- CSRs—Imaging Parameters: 800h-888h andLUT: 1A40h – 1A44h (IIDC 1.31)

8.12 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.

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

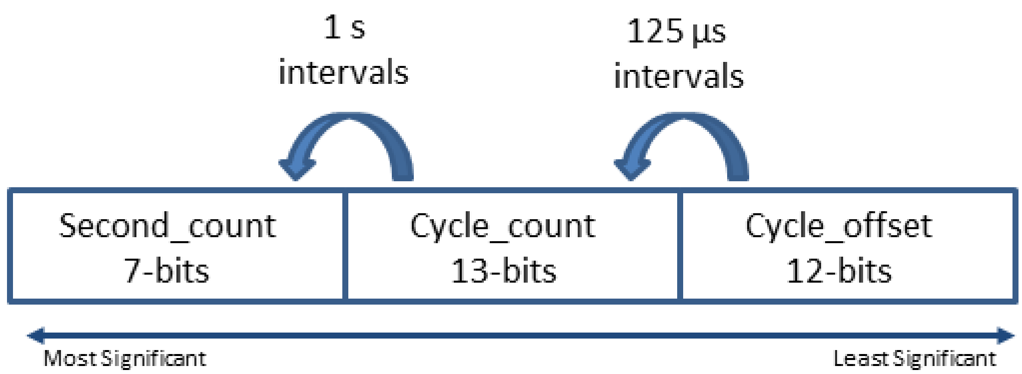
For monochrome cameras, white balance is still included, but no valid data is provided.

To access embedded information:

- CSRs—FRAME_INFO: 12F8h

Interpreting Timestamp information

The 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.

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.

8.13 White Balance

White balance is applicable to color models only.

The Chameleon 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 Automatic and One Push white balance. One use of Automatic and One Push 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 Automatic and One Push results in a similar color balance between the cameras.

One Push 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. One Push 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.

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

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

To adjust white balance:

- FlyCapture API—Setting White Balance Using the FlyCapture API
- CSRs— WHITE_BALANCE: 80Ch

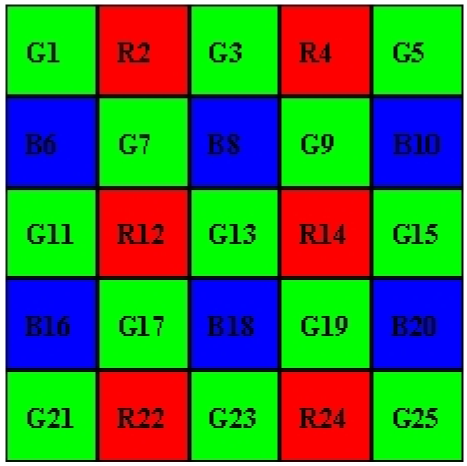
8.14 Bayer Color Processing

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 BAYER_TILE_MAPPING: 1040h.

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:

$$\begin{bmatrix} Y_{601} \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256}\begin{bmatrix} 65.738 & 129.057 & 25.064 \\ -37.945 & -74.494 & 112.439 \\ 112.439 & -94.154 & -18.285 \end{bmatrix}\begin{bmatrix} R_{255} \\ G_{255} \\ B_{255} \end{bmatrix}$$

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:

$$Y = \frac{R}{4} + \frac{G}{2} + \frac{B}{4}$$

To control Bayer color processing:

- FlyCapture API—Accessing Raw Bayer Data using FlyCapture
- CSRs—BAYER_TILE_MAPPING: 1040h

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 33
Writing color processing software and color interpolation algorithms	Knowledge Base Article 37
How is color processing performed on my camera's images?	Knowledge Base Article 89

8.15 Hue

Hue is applicable to color models only.

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

To adjust hue use:

- FlyCapture API—Setting Hue Using the FlyCapture API
- CSRs—Imaging Parameters: 800h-888h

8.16 Saturation

Saturation is applicable to color models only.

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

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

To adjust saturation use:

- FlyCapture API—Setting Saturation Using the FlyCapture API
- CSRs—Imaging Parameters: 800h-888h

9 Troubleshooting

9.1 Support

Point Grey Research endeavors to provide the highest level of technical support possible to our customers. Most support resources can be accessed through the Point Grey Product Support page.

Creating a Customer Login Account

The first step in accessing our technical support resources is to obtain a Customer Login Account. This requires a valid name and e-mail address. To apply for a Customer Login Account go to the Product Downloads page.

Knowledge Base

Our Knowledge Base contains answers to some of the most common support questions. It is constantly updated, expanded, and refined to ensure that our customers have access to the latest information.

Product Downloads

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

Contacting Technical Support

Before contacting Technical Support, have you:

1. Read the product documentation and user manual?
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:

Initialize—This allows the user to reset the camera to its initial state and default settings.

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.

Video Mode Error—This reports any camera configuration errors. If an error has occurred, no image data will be sent by the camera.

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

- CSRs—Camera Diagnostics

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 on	Receiving power and successful camera initialization
Steady on and very bright	Acquiring/transmitting images
Flashing bright, then brighter	Camera registers being accessed (no image acquisition)
Steady or slow flashing on and off	Indicates possible camera problem

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.

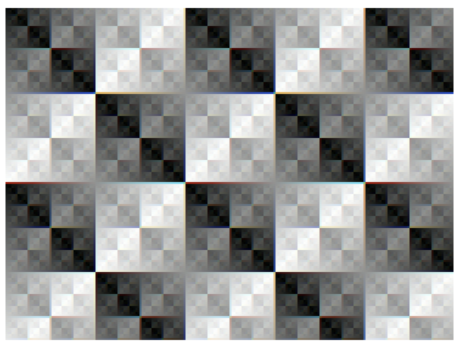


Figure 9.1: Test Pattern Sample Image

To use test pattern:

- CSRs—TEST_PATTERN: 104Ch

9.5 Blemish Pixel Artifacts

Cosmic radiation may cause random pixels to generate a permanently high charge, resulting in a permanently lit, or 'glowing,' appearance. Point Grey 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).

9.5.1 Pixel Defect Correction

Point Grey 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.

Pixel correction is not done in any of the binning modes.

Related Knowledge Base Articles

Title	Article
How Point Grey tests for white blemish pixels	Knowledge Base Article 314

To access pixel correction use:

- CSRs—PIXEL_DEFECT_CTRL: 1A60h

9.6 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.

9.6.1 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.

Related Knowledge Base Articles

Title	Article
Vertical bleeding or smearing from a saturated portion of an image	Knowledge Base Article 88

9.7 Dark Current Accumulation

Dark current refers to charge that accumulates in pixel wells in complete darkness. This effect may be especially noticeable when the camera is operating at higher temperatures.

Dark current may be minimized by reducing gain, or enabling the Min_Dark_Noise bit (bit [6]) of AUTO_SHUTTER_RANGE: 1098h. This feature can be enabled only when the camera is operating in free-running mode or trigger Mode_0.

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.parameter = 0;
mStrobe.onOff = true;
mStrobe.polarity = 1;
mStrobe.delay = 1.0f;
mStrobe.duration = 1.5f
cam.SetStrobeControl(&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.

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>FlyCapture2 SDK >Examples***

B.1 AsyncTriggerEx

The AsyncTriggerEx example program demonstrates some of the basic asynchronous trigger capabilities of compatible PGR Imaging Products.

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 PGR Imaging Product 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 Point Grey Imaging Products 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.

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 the Point Grey 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 Point Grey Research 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.

Visual Studio 2005 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_WPFshows 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

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:

numOfPackets = ceiling((rows * cols * bytesPerPixel)/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 Point Grey Digital Camera Register Reference.

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.*

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 153

B.15 MultipleCameraEx

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

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.

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

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 Isochronous Packet Format

Unlike simple register reads and writes, which are handled by asynchronous communication, the camera transmits image data using a guaranteed bandwidth mechanism known as isochronous data transmission. This section details the format and bandwidth requirements of the isochronous data broadcast by the camera. The amount of isochronous bandwidth allocated to a camera must be negotiated with the isochronous resource manager node (generally the host adapter in the PC). Every video format, mode and frame rate has a different video data format.

C.1 Isochronous Packet Format

The following table shows the format of the first 32-bits in the data field of an isochronous data block for Format 0, Format 1, Format 2, and Format 7.

Table C.1: Isochronous Data Packet Format for Format_0, Format_1 and Format_2

<table>
<tr><td>0-7</td><td>8-15</td><td colspan="2">16-23</td><td colspan="2">24-31</td></tr>
<tr><td colspan="2">data_length</td><td>tag</td><td>channel</td><td>tCode</td><td>sy</td></tr>
<tr><td colspan="2">Data Length
Number of bytes in the data field</td><td>Tag Field
set to 0</td><td>Isochronous Channel Number
programmed in the iso_channel field of the cam_sta_ctrl register</td><td>Transaction Code
set to the isochronous data block packet tCode</td><td>Synchronization Value
set to 0001h on the first isochronous data block of a frame, and set to zero on all other isochronous data blocks</td></tr>
<tr><td colspan="6">header_CRC</td></tr>
<tr><td colspan="6">Video Data Payload
contains the digital video information</td></tr>
<tr><td colspan="6">data_CRC</td></tr>
</table>

C.2 Isochronous Packet Format for Format 7

The following table shows the format of the first 32-bits in the data field of an isochronous data block.

Table C.2: Isochronous Data Packet Format for Format 7

<table>
<tr><td>0-7</td><td>8-15</td><td colspan="2">16-23</td><td colspan="2">24-31</td></tr>
<tr><td colspan="2">data_length</td><td>tag</td><td>channel</td><td>tCode</td><td>sy</td></tr>
<tr><td colspan="6">header_CRC</td></tr>
<tr><td colspan="6">Video data payload</td></tr>
<tr><td colspan="6">data_CRC</td></tr>
</table>

data_length – the number of bytes in the data field.

tag – (tag field) shall be set to 0

channel – isochronous channel number, as programmed in the iso_channel field of the cam_sta_ctrl register

tCode – (transaction code) shall be set to the isochronous data block packet tCode.

sy – (synchronization value) shall be set to 0001h on the first isochronous data block of a frame, and shall be set to zero on all other isochronous data blocks.

Video data payload – shall contain the digital video information.

C.3 Isochronous Bandwidth Requirements: Format 0, Format 1, and Format 2

The amount of isochronous bandwidth required to transmit images from the camera is dependent on the format and frame rate. The following table describes the bandwidth requirements for each available format and frame rate. Each entry in the table indicates the required bandwidth in number of lines, pixels and 32-bits per isochronous period.

Bandwidth requirements for Format 7 are negotiated with the camera at runtime.

Format_0

Mode	Video Format	240fps	120fps	60fps	30fps	15fps	7.5fps	3.75fps	1.875fps
0	160x120 YUV(4:4:4) 24bit/pixel	4H 640p 480q	2H 320p 240q	1H 160p 120q	1/2H 80p 60q	1/4H 40p 30q	1/8H 20p 15q		
1	320x240 YUV(4:2:2) 16bit/pixel	8)8H 2560p 1280q	4)4H 1280p 640q	2H 640p 320q	1H 320p 160q	1/2H 160p 80q	1/4H 80p 40q	1/8H 40p 20q	1/16H 20p 10q
2	640x480 YUV(4:1:1) 12bit/pixel	16)16H 10240p 3840q	8)8H 5120p 1920q	4)4H 2560p 960q	2)2H 1280p 480q	1H 640p 240q	1/2H 320p 120q	1/4H 160p 60q	1/8H 80p 30q
3	640x480 YUV(4:2:2) 16bit/pixel	32)16H 10240p 5120q	16)8H 5120p 2560q	8)4H 2560p 1280q	4)2H 1280p 640q	2)1H 640p 320q	1/2H 320p 160q	1/4H 160p 80q	1/8H 80p 40q
4	640x480 RGB 24bit/pixel	32)16H 10240p 7680q	16)8H 5120p 3840q	8)4H 2560p 1920q	4)2H 1280p 960q	2)1H 640p 480q	1/2H 320p 240q	1/4H 160p 120q	1/8H 80p 60q
5	640x480 Y (Mono) 8bit/pixel	16)16H 10240p 2560q	8)8H 5120p 1280q	4)4H 2560p 640q	2)2H 1280p 320	1H 640p 160q	1/2H 320p 80q	1/4H 160p 40q	1/8H 80p 20q

Mode	Video Format	240fps	120fps	60fps	30fps	15fps	7.5fps	3.75fps	1.875fps
6	640x480 Y (Mono) 16bit/pixel	32)16H 10240p 5120q	16)8H 5120p 2560q	8)4H 2560p 1280q	4)2H 1280p 640q	2)1H 640p 320q	1/2H 320p 160q	1/4H 160p 80q	1/8H 80p 40q
7	Reserved								

Format_1

Mode	Video Format	240fps	120fps	60fps	30fps	15fps	7.5fps	3.75fps	1.875fps
0	800*600 YUV(4:2:2) 16bit/pixel	32)20H 16000p 8000q	16)10H 8000p 4000q	8)5H 4000p 2000q	4)5/2H 2000p 1000q	2)5/4H 1000p 500q	5/8H 500p 250q	5/16H 250p 125q	
1	800x600 RGB 24bit/pixel		32)10H 8000p 600q	16)5H 4000p 3000q	8)5/2H 2000p 1500q	4)5/4H 1000p 750q	2)5/8H 500p 375q		
2	800x600 Y (Mono) 8bit/pixel	16)20H 16000p 4000q	8)10H 8000p 2000q	4)5H 4000p 1000q	2)5/2H 2000p 500q	5/4H 1000p 250q	5/8H 500p 125q		
3	1024x768 YUV (4:2:2) 16bit/pixel		32)12H 12288p 6144q	16)6H 6144p 3072q	8)3H 3072p 1536q	4)3/2H 1536p 768q	2)3/4H 768p 384q	3/8H 384p 192q	3/16H 192p 96q
4	1024x768 RGB 24bit/pixel			32)6H 6144p 4608q	16)3H 3072p 2304q	8)3/2H 1536p 1152q	4)3/4H 768p 576q	2)3/8H 384p 288q	3/16 192p 144q
5	1024x768 Y (Mono) 8bit/pixel	32)24H 24576p 6144q	16)12H 12288p 3072q	8)6H 6144p 1536q	4)3H 3072p 768q	2)3/2H 1536p 384q	3/4H 768p 192q	3/8H 384p 96q	3/16H 192p 48q
6	800x600 Y (Mono16) 16bit/pixel	32)20H 16000p 8000q	16)10H 8000p 4000q	8)5H) 4000p 2000q	4)5/2H 2000p 1000q	2)5/4H 1000p 500q	5/8H 500p 250q	5/16H 250p 125q	
7	1024x768 Y (Mono16) 16bit/pixel		32)12H 12288p 6144q	16)6H 6144p 3072q	8)3H 3072p 1536q	4)3/2H 1536p 768q	2)3/4H 768p 384q	3/8H 384p 192q	3/16H 192p 96q

Format_2

Mode	Video Format	120fps	60fps	30fps	15fps	7.5fps	3.75fps	1.875fps
0	1280x960 YUV(4:2:2) 16bit/pixel		32)8H 10240p 5120q	16)4H 5120p 2560q	8)2H 2560p 1280q	4)1H 1280p 640q	2)1/2H 640p 320q	1/4H 320p 160q
1	1280x960 RGB 24bit/pixel		32)8H 10240p 7680q	16)4H 5120p 3840q	8)2H 2560p 1920q	4)1H 1280p 960q	2)1/2H 640p 480q	1/4H 320p 240q
2	1280x960 Y (Mono) 8bit/pixel	32)16H 20480p 5120q	16)8H 10240p 2560q	8)4H 5120p 1280q	4)2H 2560p 640q	2)1H 1280p 320q	1/2H 640p 160q	1/4H 320p 80q
3	1600x1200 YUV(4:2:2) 16bit/pixel		32)10H 16000p 8000q	16)5H 8000p 4000q	8)5/2H 4000p 2000q	4)5/4H 2000p 1000q	2)5/8H 1000p 500q	5/16H 500p 250q
4	1600x1200 RGB 24bit/pixel			32)5H 8000p 6000q	16)5/2H 4000p 3000q	8)5/4H 2000p 1500q	4)5/8H 1000p 750q	2)5/16H 500p 375q
5	1600x1200 Y (Mono) 8bit/pixel	32)20H 32000p 8000q	16)10H 16000p 4000q	8)5H 8000p 2000q	4)5/2H 4000p 1000q	2)5/4H 2000p 500q	5/8H 1000p 250q	5/16H 500p 125q
6	1280x960 Y (Mono16) 16bit/pixel		32)8H 10240p 5120q	16)4H 5120p 2560q	8)2H 2560p 1280q	4)1H 1280p 640q	2)1/2H 640p 320q	1/4H 320p 160q
7	1600x1200 Y(Mono16) 16bit/pixel		32)10H 16000p 8000q	16)5H 8000p 4000qH	8)5/2H 4000p 2000q	4)5/4H 2000p 1000q	2)5/8H 1000p 500q	5/16H 500p 250q

[--H – Lines/Packet]

[--p – Pixels/Packet]

[--q – 32-bits/Packet

2) : required S200 data rate

4) : required S400 data rate

8) : required S800 data rate

16) : required S1600 data rate

32) : required S3200 data rate

D 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.

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

D.1 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.

Not all features implement all modes.

Table D.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.

D.2 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.

D.3 Register Memory Map

The camera uses a 64-bit fixed addressing model. The upper 10 bits show the Bus ID, and the next six bits show the Node ID. The next 20 bits must be 1 (FFFF Fh).

Address	Register Name	Description
FFFF F000 0000h	Base address	
FFFF F000 0400h	Config ROM	
FFFF F0F0 0000h	Base address for all camera control command registers	
The following register addresses are offset from the base address, FFFF F0F0 0000h.		
000h	INITIALIZE	Camera initialize register
100h 180h	V_FORMAT_INQ V_MODE_INQ_X	Inquiry register for video format Inquiry register for video mode
200h	V_RATE_INQ_y_X	Inquiry register for video frame rate
300h	Reserved	
400h	BASIC_FUNC_INQ FEATURE_HI_INQ FEATURE_LO_INQ	Inquiry register for feature presence
500h	*Feature_Name*_INQ	Inquiry register for feature elements
600h 640h	CAM_STA_CTRL	Status and control register for camera Feature control error status register
700h	ABS_CSR_HI_INQ_x	Inquiry register for Absolute value CSR offset address
800h	Feature_Name	Status and control register for feature

The FlyCapture API library has function calls to get and set camera register values. These function calls automatically take into account the base address. For example, to get the 32-bit value of the SHUTTER register at 0xFFFF F0F0 081C:

FlyCapture v1.x:

```
flycaptureGetCameraRegister(context, 0x81C, &ulValue);
flycaptureSetCameraRegister(context, 0x81C, ulValue);
```

FlyCapture v2.x (assuming a `camera` object named `cam`):

```
cam.ReadRegister(0x81C, &regVal);
cam.WriteRegister(0x81C, regVal, broadcast=false);
```

Broadcast is only available for FlyCapture2 and FireWire cameras. FireWire has the ability to write to multiple cameras at the same time.

D.4 Config ROM

D.4.1 Root Directory

	Offset	Bit	Description
Bus Info Block	400h	[0-7]	04h
		[8-15]	crc_length
		[16-31]	rom_crc_value
	404h	[0-7]	31h
		[8-15]	33h
		[16-23]	39h
		[24-31]	34h
	408h	[0-3]	0010 (binary)
		[4-7]	Reserved
		[8-15]	FFh
		[16-19]	max_rec
		[20]	Reserved
		[21-23]	mxrom
		[24-31]	chip_id_hi
	40Ch	[0-23]	node_vendor_id
		[24-31]	chip_id_hi
	410h	[0-31]	chip_id_lo
Root Directory	414h	[0-15]	0004h
		[16-31]	CRC
	418h	[0-7]	03h
		[8-31]	module_vendor_id
	41Ch	[0-7]	0Ch
		[8-15]	Reserved
		[16-31]	1000001111000000 (binary)
	420h	[0-7]	8Dh
		[8-31]	indirect_offset
	424h	[0-7]	D1h
		[8-31]	unit_directory_offset

D.4.2 Unit Directory

Offset	Bit	Description
0000h	[0-15]	0003h
	[16-31]	CRC

Offset	Bit	Description
0004h	[0-7]	12h
	[8-31]	unit_spec_ID (=0x00A02D)
0008h	[0-7]	13h
	[8-31]	unit_sw_version (=0x000102)
000Ch	[0-7]	D4h
	[8-31]	unit dependent directory offset

D.4.3 Unit Dependent Info

Offset	Bit	Description
0000h	[0-15]	unit_dep_info_length
	[16-31]	CRC
0004h	[0-7]	40h
	[8-31]	command_regs_base 32-bit offset from the base address of initial register space of the base address of the command registers
0008h	[0-7]	81h
	[8-31]	vendor_name_leaf The number of 32-bits from the address of the vendor_name_leaf entry to the address of the vendor_name leaf containing an ASCII representation of the vendor name of this node
000Ch	[0-7]	82h
	[8-31]	model_name_leaf The number of 32-bits from the address of the model_name_leaf entry to the address of the model_name leaf containing an ASCII representation of the model name of this node
0010h	[0-7]	38h
	[8-31]	unit_sub_sw_version the sub version information of this unit unit_sub_sw_version = 0x000000h or unspecified for IIDC v1.30 unit_sub_sw_version = 0x000010h for IIDC v1.31 unit_sub_sw_version = 0x000020h for IIDC v1.32
0014h	[0-7]	39h
	[8-31]	Reserved
0018h	[0-7]	3Ah
	[8-31]	Reserved
001Ch	[0-7]	3Bh
	[8-31]	Reserved
0020h	[0-7]	3Ch
	[8-31]	vendor_unique_info_0
0024h	[0-7]	3Dh
	[8-31]	vendor_unique_info_1
0028h	[0-7]	3Eh
	[8-31]	vendor_unique_info_2

Offset	Bit	Description
002Ch	[0-7]	3Fh
	[8-31]	vendor_unique_info_3

D.5 Calculating Base Register Addresses using 32-bit Offsets

The addresses for many CSRs, such as those that provide control over absolute values, custom video modes, PIO, SIO and strobe output, can vary between cameras. In order to provide a common mechanism across camera models for determining the location of these CSRs relative to the base address, there are fixed locations for inquiry registers that contain offsets, or pointers, to the actual offsets.

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)*

For example, the Absolute Value CSRs provide minimum, maximum and current real-world values for camera properties such as gain, shutter, etc., as described in *Absolute Value Registers* (on the next page). To determine the location of the shutter absolute value registers (code snippets use function calls included in the FlyCapture SDK, and assume a `Camera` object `cam`):

1. Read the ABS_CSR_HI_INQ_7 register 71Ch to obtain the 32-bit offset for the absolute value CSR for shutter.

```
unsigned int ulValue;
cam.ReadRegister(0x71C, &ulValue);
```

2. The ulValue is a 32-bit offset, so multiply by 4 to get the actual offset.

```
ulValue = ulValue * 4; // ulValue == 0x3C0244, actual offset == 0xF00910
```

3. The actual offset 0xF00910 represents the offset from the base address 0xFFFF Fxxx xxxx. Since the PGR FlyCapture API automatically takes into account the base offset 0xFFFF F0F0 0000, the actual offset in this example would be 0x910.

```
ulValue = ulValue & 0xFFFF;
```

D.6 Absolute Value Registers

Many Point Grey cameras implement "absolute" modes for various camera settings that report real-world values, such as shutter time in seconds (s) and gain value in decibels (dB). Using these absolute values is easier and more efficient than applying complex conversion formulas to the information in the *Value* field of the associated Control and Status Register. A relative value does not always translate to the same absolute value. Two properties that can affect this relationship are pixel clock frequency and horizontal line frequency. These properties are, in turn, affected by such properties as resolution, frame rate, region of interest (ROI) size and position, and packet size. Additionally, conversion formulas can change between firmware versions. Point Grey therefore recommends using absolute value registers, where possible, to determine camera values.

D.6.1 Setting Absolute Value Register Values

For absolute values to be used, the associated feature CSR must be set to use absolute values.

Field	Bit	Description
Abs_Control	[1]	Absolute value control 0: Control with the value in the Value field 1: Control with the value in the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.

In the FlyCapture API, this can also be done by setting the `absControl` member of the of the desired property structure to true.

D.6.2 Absolute Value Offset Addresses

The following set of registers indicates the locations of the absolute value registers. Not all cameras use all registers.

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)*

32-bit Offsets for Absolute Value Registers

Offset	Name	Bit	Description
700h	ABS_CSR_HI_INQ_0	[0..31]	Brightness
704h	ABS_CSR_HI_INQ_1	[0..31]	Auto Exposure
708h	ABS_CSR_HI_INQ_2	[0..31]	Sharpness
710h	ABS_CSR_HI_INQ_4	[0..31]	Hue
714h	ABS_CSR_HI_INQ_5	[0..31]	Saturation
718h	ABS_CSR_HI_INQ_6	[0..31]	Gamma
71Ch	ABS_CSR_HI_INQ_7	[0..31]	Shutter
720h	ABS_CSR_HI_INQ_8	[0..31]	Gain
724h	ABS_CSR_HI_INQ_9	[0..31]	Iris
734h	ABS_CSR_HI_INQ_13	[0..31]	Trigger Delay

Offset	Name	Bit	Description
73Ch	ABS_CSR_HI_INQ_15	[0..31]	Frame Rate
7C4h	ABS_CSR_LO_INQ_1	[0..31]	Pan
7C8h	ABS_CSR_LO_INQ_2	[0..31]	Tilt

Each set of absolute value CSRs consists of three registers as follows:

Offset	Name	Field	Bit	Description
Base + 000h	Absolute Value	Min_Value	[0-31]	Minimum value for this feature. Read only.
Base + 004h		Max_Value	[0-31]	Maximum value for this feature. Read only.
Base + 008h		Value	[0-31]	Current value of this feature.

For example:

Offset	Name	Field	Bit	Description
704h	ABS_CSR_HI_INQ_1		[0..31]	Auto Exposure.
Base + 0h	ABS_VAL_AUTO_EXPOSURE	Min_Value	[0-31]	Min auto exposure value.
Base + 4h		Max_Value	[0-31]	Max auto exposure value.
Base + 8h		Value	[0-31]	Current auto exposure value.

D.6.3 Units of Value for Absolute Value CSR Registers

The following tables describe the real-world units that are used for the absolute value registers. Each value is either Absolute (value is an absolute value) or Relative (value is an absolute value, but the reference is system dependent).

Feature	Function	Unit	Unit Description	Reference point	Value Type
Brightness	Black level offset	%		----	Absolute
Auto Exposure	Auto Exposure	EV	exposure value	0	Relative
Sharpness	Sharpness				
Hue	Hue	deg	degree	0	Relative
Saturation	Saturation	%		100	Relative
Gamma					
Shutter	Integration time	s	seconds	----	Absolute
Gain	Circuit gain	dB	decibel	0	Relative
Iris	Iris	F	F number	----	Absolute
Trigger_Delay	Trigger Delay	S	seconds	----	Absolute
Frame_Rate	Frame rate	fps	frames per second	----	Absolute
Pan	Pan				
Tilt	Tilt				

D.6.4 Determining Absolute Value Register Values

The Absolute Value CSRs store 32-bit floating-point values with IEEE/REAL*4 format. To programmatically determine the floating point equivalents of the minimum, maximum and current hexadecimal values for a property such as shutter, using the FlyCapture SDK:

1. Read the ABS_CSR_HI_INQ_7 register 71Ch to obtain the 32-bit offset for the absolute value CSR for shutter.

```
cam.ReadRegister(context, 0x71C, &ulValue);
```

2. The ulValue is a 32-bit offset, so multiply by 4 to get the actual offset.

```
ulValue = ulValue * 4; // ulValue == 0x3C0244, actual offset == 0xF00910
```

This offset represents the offset from the base address 0xFFFF Fxxx xxxx. Since the PGR FlyCapture API automatically takes into account the base offset 0xFFFF F0F0 0000, the actual offset in this example would be 0x910.

3. Use the offset obtained to read the min, max and current absolute values and convert the 32-bit hexadecimal values to floating point.

```
// declare a union of a floating point and unsigned long
typedef union _AbsValueConversion
{
         unsigned long ulValue;
         float fValue;
} AbsValueConversion;
float    fMinShutter,    fMaxShutter,    fCurShutter;AbsValueConversion
minShutter, maxShutter, curShutter;
// read the 32-bit hex value into the unsigned long member
cam.ReadRegister(context, 0x910, &minShutter.ulValue );
cam.ReadRegister(context, 0x914, &maxShutter.ulValue );
cam.ReadRegister(context, 0x918, &curShutter.ulValue );
fMinShutter = minShutter.fValue;
fMaxShutter = maxShutter.fValue;
fCurShutter = curShutter.fValue;
```

To get and set absolute values using the FlyCapture SDK, use the `GetProperty` *and* `SetProperty` *functions to get or set the* `absValue` *member of the* `Property` *struct. Refer to the FlyCapture SDK Help for function definitions.*

D.7 Inquiry Registers

D.7.1 Basic Functions Inquiry Registers

The following registers show which basic functions are implemented on the camera.

(Bit values = 0: Not Available, 1: Available)

Format:

Offset	Name	Field	Bit	Description
400h	BASIC_FUNC_INQ	Advanced_Feature_Inq	[0]	Inquiry for advanced feature. (Vendor Unique Features)
		Vmode_Error_Status_Inq	[1]	Inquiry for existence of Vmode_Error_Status register
		Feature_Control_Error_Status_Inq	[2]	Inquiry for existence of Feature_Control_Error_ Status register
		Opt_Func_CSR_Inq	[3]	Inquiry for optional function CSR.
			[4-7]	Reserved
		1394.b_mode_Capability	[8]	Inquiry for 1394.b mode capability
			[9-15]	Reserved
		Cam_Power_Cntl	[16]	Camera process power ON/OFF capability
			[17-18]	Reserved
		One_Shot_Inq	[19]	One shot transmission capability
		Multi_Shot_Inq	[20]	Multi shot transmission capability
		Retransmit_Inq	[21]	Retransmit latest image capability (One_ shot/Retransmit)
		Image_Buffer_Inq	[22]	Image buffer capability (Multi_shot/Image_Buffer)
			[23-27]	Reserved
		Memory_Channel	[28-31]	Maximum memory channel number (N) Memory channel 0 = Factory setting memory 1 = Memory Ch 1 2 = Memory Ch 2 : N= Memory Ch N If 0000, user memory is not available.

D.7.2 Feature Presence Inquiry Registers

The following registers show the presence of the camera features or optional functions implemented on the camera.

(Bit values = 0: Not Available, 1: Available)

Format:

Offset	Name	Field	Bit	Description
404h	Feature_Hi_Inq	Brightness	[0]	Brightness Control
		Auto_Exposure	[1]	Auto Exposure Control
		Sharpness	[2]	Sharpness Control
		White_Balance	[3]	White Balance Control
		Hue	[4]	Hue Control
		Saturation	[5]	Saturation Control
		Gamma	[6]	Gamma Control
		Shutter	[7]	Shutter Speed Control
		Gain	[8]	Gain Control
		Iris	[9]	IRIS Control
		Focus	[10]	Focus Control
		Temperature	[11]	Temperature Control
		Trigger	[12]	Trigger Control
		Trigger_Delay	[13	Trigger Delay Control
		White_Shading	[14]	White Shading Compensation Control
		Frame_Rate	[15]	Frame rate prioritize control
			[16-31]	Reserved
408h	Feature_Lo_Inq	Zoom	[0]	Zoom Control
		Pan	[1]	Pan Control
		Tilt	[2]	Tilt Control
		Optical Filter	[3]	Optical Filter Control
			[4-15]	Reserved
		Capture_Size	[16]	Capture image size for Format_6
		Capture_Quality	[17]	Capture image quality for Format_6
			[18-31]	Reserved
40Ch	Opt_Function_Inq	-	[0]	Reserved
		PIO	[1]	Parallel input/output control
		SIO	[2]	Serial Input/output control
		Strobe_Output	[3]	Strobe signal output
		Lookup_Table	[4]	Lookup table control
		-	[5-31]	Reserved
410h-47Fh	Reserved			
480h	Advanced_Feature_Inq	Advanced_ Feature_Quadlet_ Offset	[0-31]	32-bit offset of the advanced feature CSRs from the base address of initial register space. (Vendor unique)
484h	PIO_Control_CSR_Inq	PIO_Control_ Quadlet_Offset	[0-31]	32-bit offset of the PIO control CSRs from the base address of initial register space.
488h	SIO_Control_CSR_Inq	SIO_Control_ Quadlet_Offset	[0-31]	32-bit offset of the SIO control CSRs from the base address of initial register space.

Offset	Name	Field	Bit	Description
48Ch	Strobe_Output_CSR_Inq	Strobe_Output_ Quadlet_Offset	[0-31]	32-bit offset of the strobe output signal CSRs from the base address of initial register space.
490h	Lookup_Table_CSR_Inq	Lookup_Table_ Quadlet_Offset	[0-31]	32-bit offset of the Lookup Table CSRs from the baes address of initial register space.

D.7.3 Feature Elements Inquiry Registers

The following registers show the presence of specific features, modes and minimum and maximum values for each of the camera features or optional functions implemented by the camera.

(Bit values = 0: Not Available, 1: Available)

Offset	Name	Field	Bit	Description
500h	BRIGHTNESS_INQ	Presence_Inq	[0]	Presence of this feature
		Abs_Control_Inq	[1]	Absolute value control
			[2]	Reserved
		One_Push_Inq	[3]	One push mode (controlled automatically only once)
		ReadOut_Inq	[4]	Ability to read the value of this feature
		On_Off_Inq	[5]	Ability to switch feature ON and OFF
		Auto_Inq	[6]	Auto mode (controlled automatically)
		Manual_Inq	[7]	Manual mode (controlled by user)
		Min_Value	[8-19]	Minimum value for this feature control
		Max_Value	[20-31]	Maximum value for this feature control
504h	AUTO_EXPOSURE_INQ	Same format as the BRIGHTNESS_INQ register		
508h	SHARPNESS_INQ	Same format as the BRIGHTNESS_INQ register		
50Ch	WHITE_BALANCE_INQ	Same format as the BRIGHTNESS_INQ register		
510h	HUE_INQ	Same format as the BRIGHTNESS_INQ register		
514h	SATURATION_INQ	Same format as the BRIGHTNESS_INQ register		
518h	GAMMA_INQ	Same format as the BRIGHTNESS_INQ register		
51Ch	SHUTTER_INQ	Same format as the BRIGHTNESS_INQ register		
520h	GAIN_INQ	Same format as the BRIGHTNESS_INQ register		
524h	IRIS_INQ	Same format as the BRIGHTNESS_INQ register		
528h	FOCUS_INQ	Same format as the BRIGHTNESS_INQ register		
52Ch	TEMPERATURE_INQ	Same format as the BRIGHTNESS_INQ register		

Offset	Name	Field	Bit	Description
530h	TRIGGER_INQ	Presence_Inq	[0]	Presence of this feature
		Abs_Control_Inq	[1]	Absolute value control
			[2-3]	Reserved
		ReadOut_Inq	[4]	Ability to read the value of this feature
		On_Off_Inq	[5]	Ability to switch feature ON and OFF
		Polarity_Inq	[6]	Ability to change trigger input polarity
		Value_Read_Inq	[7]	Ability to read raw trigger input
		Trigger_Source0_Inq	[8]	Presence of Trigger Source 0 ID=0
		Trigger_Source1_Inq	[9]	Presence of Trigger Source 1 ID=1
		Trigger_Source2_Inq	[10]	Presence of Trigger Source 2 ID=2
		Trigger_Source3_Inq	[11]	Presence of Trigger Source 3 ID=3
			[12-14]	Reserved
		Software_Trigger_Inq	[15]	Presence of Software Trigger ID=7
		Trigger_Mode0_Inq	[16]	Presence of Trigger Mode 0
		Trigger_Mode1_Inq	[17]	Presence of Trigger Mode 1
		Trigger_Mode2_Inq	[18]	Presence of Trigger Mode 2
		Trigger_Mode3_Inq	[19]	Presence of Trigger Mode 3
		Trigger_Mode4_Inq	[20]	Presence of Trigger Mode 4
		Trigger_Mode5_Inq	[21]	Presence of Trigger Mode 5
			[22-29]	Reserved
		Trigger_Mode14_Inq	[30]	Presence of Trigger Mode 14 (Vendor unique trigger mode 0)
		Trigger_Mode15_Inq	[31]	Presence of Trigger Mode 15 (Vendor unique trigger mode 1)
534h	TRIGGER_DLY_INQ	Presence_Inq	[0]	Presence of this feature
		Abs_Control_Inq	[1]	Absolute value control
			[2]	Reserved
		One_Push_Inq	[3]	One push mode (controlled automatically only once)
		ReadOut_Inq	[4]	Ability to read the value of this feature
		On_Off_Inq	[5]	Ability to switch feature ON and OFF
			[6-7]	Reserved
		Min_Value	[8-19]	Minimum value for this feature control
		Max_Value	[20-31]	Maximum value for this feature control
538h	WHITE_SHD_INQ	Same format as the BRIGHTNESS_INQ register		
53Ch	FRAME_RATE_INQ	Same format as the BRIGHTNESS_INQ register		
540h : 57Ch	Reserved for other FEATURE_HI_INQ			
580h	ZOOM_INQ	Same format as the BRIGHTNESS_INQ register		

Offset	Name	Field	Bit	Description
584h	PAN_INQ	Same format as the BRIGHTNESS_INQ register		
588h	TILT_INQ	Same format as the BRIGHTNESS_INQ register		
58Ch	OPTICAL_FILTER_INQ	Same format as the BRIGHTNESS_INQ register		

D.7.4 Video Format Inquiry Registers

The following registers may be used to determine the video formats that are available with the camera.

(Bit values = 0: Not Available, 1: Available)

Format:

Offset	Name	Field	Bit	Description
100h	V_FORMAT_INQ	Format_0	[0]	VGA non-compressed format (160x120 through 640x480)
		Format_1	[1]	Super VGA non-compressed format (1) (800x600 through 1024x768)
		Format_2	[2]	Super VGA non-compressed format (2) (1280x960 through 1600x1200)
		Format_x	[3-5]	Reserved for other formats
		Format_6	[6]	Still Image Format
		Format_7	[7]	Partial Image Size Format
			[8-31]	Reserved

D.7.5 Video Mode Inquiry Registers

The following registers may be used to determine the video modes that are available with the camera.

(Bit values = 0: Not Available, 1: Available)

Format:

Offset	Name	Field	Bit	Description
180h	V_MODE_INQ_O (Format 0)	Mode_0	[0]	160 x 120 YUV(4:4:4) Mode (24 bits/pixel)
		Mode_1	[1]	320 x 240 YUV(4:2:2) Mode (16 bits/pixel)
		Mode_2	[2]	640 x 480 YUV(4:1:1) Mode (12 bits/pixel)
		Mode_3	[3]	640 x 480 YUV(4:2:2) Mode (16 bits/pixel)
		Mode_4	[4]	640 x 480 RGB Mode (24 bits/pixel)
		Mode_5	[5]	640 x 480 Y8 (Mono) Mode (8 bits/pixel)
		Mode_6	[6]	640 x 480 Y16 (Mono16) Mode (16 bits/pixel)
			[7-31]	Reserved

Offset	Name	Field	Bit	Description
184h	V_MODE_INQ_1 (Format 1)	Mode_0	[0]	800 x 600 YUV(4:2:2) Mode (16 bits/pixel)
		Mode_1	[1]	800 x 600 RGB Mode (24 bits/pixel)
		Mode_2	[2]	800 x 600 Y (Mono) Mode (8 bits/pixel)
		Mode_3	[3]	1024 x 768 YUV(4:2:2) Mode (16 bits/pixel)
		Mode_4	[4]	1024 x 768 RGB Mode (24 bits/pixel)
		Mode_5	[5]	1024 x 768 Y (Mono) Mode (8 bits/pixel)
		Mode_6	[6]	800 x 600 Y (Mono16) Mode (16 bits/pixel)
		Mode_7	[7]	1024 x 768 Y (Mono16) Mode (16 bits/pixel)
			[8-31]	Reserved
188h	V_MODE_INQ_2 (Format 2)	Mode_0	[0]	1280 x 960 YUV(4:2:2) Mode (16 bits/pixel)
		Mode_1	[1]	1280 x 960 RGB Mode (24 bits/pixel)
		Mode_2	[2]	1280 x 960 Y (Mono) Mode (8 bits/pixel)
		Mode_3	[3]	1600 x 1200 YUV(4:2:2) Mode (16 bits/pixel)
		Mode_4	[4]	1600 x 1200 RGB Mode (24 bits/pixel)
		Mode_5	[5]	1600 x 1200 Y (Mono) Mode (8 bits/pixel)
		Mode_6	[6]	1280 x 960 Y (Mono16) Mode (16 bits/pixel)
		Mode_7	[7]	1600 x 1200 Y (Mono16) Mode (16 bits/pixel)
			[8-31]	Reserved
18Ch : 197h	Reserved			
19Ch	V_MODE_INQ_7 (Format 7)	Mode_0	[0]	Format 7 Mode 0
		Mode_1	[1]	Format 7 Mode 1
		Mode_2	[2]	Format 7 Mode 2
		Mode_3	[3]	Format 7 Mode 3
		Mode_4	[4]	Format 7 Mode 4
		Mode_5	[5]	Format 7 Mode 5
		Mode_6	[6]	Format 7 Mode 6
		Mode_7	[7]	Format 7 Mode 7
			[8-31]	Reserved

D.7.6 Video Frame Rate Inquiry Registers

This set of registers allows the user to query the available frame rates for all Formats and Modes.

(Bit values = 0: Not Available, 1: Available)

Offset	Name	Field	Bit	Description
200h	V_RATE_INQ_0_0 (Format 0, Mode 0)	FrameRate_0	[0]	Reserved
		FrameRate_1	[1]	Reserved
		FrameRate_2	[2]	7.5 fps
		FrameRate_3	[3]	15 fps
		FrameRate_4	[4]	30 fps
		FrameRate_5	[5]	60 fps
		FrameRate_6	[6]	120 fps
		FrameRate_7	[7]	240 fps
			[8-31]	Reserved
204h	V_RATE_INQ_0_1 (Format 0, Mode 1)	FrameRate_0	[0]	1.875 fps
		FrameRate_1	[1]	3.75 fps
		FrameRate_2	[2]	7.5 fps
		FrameRate_3	[3]	15 fps
		FrameRate_4	[4]	30 fps
		FrameRate_5	[5]	60 fps
		FrameRate_6	[6]	120 fps
		FrameRate_7	[7]	240 fps
			[8-31]	Reserved
208h	V_RATE_INQ_0_2 (Format 0, Mode 2)	Same format as V_RATE_INQ_0_1 Register (Format 0, Mode 1)		
20Ch	V_RATE_INQ_0_3 (Format 0, Mode 3)	Same format as V_RATE_INQ_0_1 Register (Format 0, Mode 1)		
210h	V_RATE_INQ_0_4 (Format 0, Mode 4)	Same format as V_RATE_INQ_0_1 Register (Format 0, Mode 1)		
214h	V_RATE_INQ_0_5 (Format 0, Mode 5)	Same format as V_RATE_INQ_0_1 Register (Format 0, Mode 1)		
218h	V_RATE_INQ_0_6 (Format 0, Mode 6)	Same format as V_RATE_INQ_0_1 Register (Format 0, Mode 1)		
21Ch : 21Fh	Reserved			
220h	V_RATE_INQ_1_0 (Format 1, Mode 0)	FrameRate_0	[0]	Reserved
		FrameRate_1	[1]	3.75 fps
		FrameRate_2	[2]	7.5 fps
		FrameRate_3	[3]	15 fps
		FrameRate_4	[4]	30 fps
		FrameRate_5	[5]	60 fps
		FrameRate_6	[6]	120 fps
		FrameRate_7	[7]	240 fps
			[8-31]	Reserved

Offset	Name	Field	Bit	Description
224h	V_RATE_INQ_1_1 (Format 1, Mode 1)	Same format as V_RATE_INQ_0_0 Register (Format 0, Mode 0)		
228h	V_RATE_INQ_1_2 (Format 1, Mode 2)	Same format as V_RATE_INQ_0_0 Register (Format 0, Mode 0)		
22Ch	V_RATE_INQ_1_3 (Format 1, Mode 3)	FrameRate_0	[0]	1.875 fps
		FrameRate_1	[1]	3.75 fps
		FrameRate_2	[2]	7.5 fps
		FrameRate_3	[3]	15 fps
		FrameRate_4	[4]	30 fps
		FrameRate_5	[5]	60 fps
		FrameRate_6	[6]	120 fps
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
230h	V_RATE_INQ_1_4 (Format 1, Mode 4)	FrameRate_0	[0]	1.875 fps
		FrameRate_1	[1]	3.75 fps
		FrameRate_2	[2]	7.5 fps
		FrameRate_3	[3]	15 fps
		FrameRate_4	[4]	30 fps
		FrameRate_5	[5]	60 fps
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
234h	V_RATE_INQ_1_5 (Format 1, Mode 5)	Same format as V_RATE_INQ_0_1 Register (Format 0, Mode 1)		
238h	V_RATE_INQ_1_6 (Format 1, Mode 6)	Same format as V_RATE_INQ_1_0 register (Format 1, Mode 0)		
23Ch	V_RATE_INQ_1_7 (Format 1, Mode 7)	Same format as V_RATE_INQ_1_3 register (Format 1, Mode 3)		
240h	V_RATE_INQ_2_0 (Format 2, Mode 0)	Same format as V_RATE_INQ_1_4 register (Format 1, Mode 4)		
244h	V_RATE_INQ_2_1 (Format 2, Mode 1)	Same format as V_RATE_INQ_1_4 register (Format 1, Mode 4)		
248h	V_RATE_INQ_2_2 (Format 2, Mode 2)	Same format as V_RATE_INQ_1_3 register (Format 1, Mode 3)		
24Ch	V_RATE_INQ_2_3 (Format 2, Mode 3)	Same format as V_RATE_INQ_1_4 register (Format 1, Mode 4)		

Offset	Name	Field	Bit	Description
250h	V_RATE_INQ_2_4 (Format 2, Mode 4)	FrameRate_0	[0]	1.875 fps
		FrameRate_1	[1]	3.75 fps
		FrameRate_2	[2]	7.5 fps
		FrameRate_3	[3]	15 fps
		FrameRate_4	[4]	30 fps
		FrameRate_5	[5]	Reserved
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
254h	V_RATE_INQ_2_5 (Format 2, Mode 5)	Same format as V_RATE_INQ_1_3 register (Format 1, Mode 3)		
258h	V_RATE_INQ_2_6 (Format 2, Mode 6)	Same format as V_RATE_INQ_1_4 register (Format 1, Mode 4)		
25Ch	V_RATE_INQ_2_7 (Format 2, Mode 7)	Same format as V_RATE_INQ_1_4 register (Format 1, Mode 4)		
260h : 2BFh	Reserved			
2E0h	V_CSR_INQ_7_0	Mode_0	[0-31]	CSR 32-bit offset for Format 7 Mode 0
2E4h	V_CSR_INQ_7_1	Mode_1	[0-31]	CSR 32-bit offset for Format 7 Mode 1
2E8h	V_CSR_INQ_7_2	Mode_2	[0-31]	CSR 32-bit offset for Format 7 Mode 2
2ECh	V_CSR_INQ_7_3	Mode_3	[0-31]	CSR 32-bit offset for Format 7 Mode 3
2F0h	V_CSR_INQ_7_4	Mode_4	[0-31]	CSR 32-bit offset for Format 7 Mode 4
2F4h	V_CSR_INQ_7_5	Mode_5	[0-31]	CSR 32-bit offset for Format 7 Mode 5
2F8h	V_CSR_INQ_7_6	Mode_6	[0-31]	CSR 32-bit offset for Format 7 Mode 6
2FCh	V_CSR_INQ_7_7	Mode_7	[0-31]	CSR 32-bit offset for Format 7 Mode 7
300h	V_CSR_INQ_7_8	Mode_8	[0-31]	CSR 32-bit offset for Format 7 Mode 8
304h	V_CSR_INQ_7_9	Mode_9	[0-31]	CSR 32-bit offset for Format 7 Mode 9
308h	V_CSR_INQ_7_10	Mode_10	[0-31]	CSR 32-bit offset for Format 7 Mode 10
30Ch	V_CSR_INQ_7_11	Mode_11	[0-31]	CSR 32-bit offset for Format 7 Mode 11
310h	V_CSR_INQ_7_12	Mode_12	[0-31]	CSR 32-bit offset for Format 7 Mode 12
314h	V_CSR_INQ_7_13	Mode_13	[0-31]	CSR 32-bit offset for Format 7 Mode 13
318h	V_CSR_INQ_7_14	Mode_14	[0-31]	CSR 32-bit offset for Format 7 Mode 14
31Ch	V_CSR_INQ_7_15	Mode_15	[0-31]	CSR 32-bit offset for Format 7 Mode 15
320h	V_CSR_INQ_7_16	Mode_16	[0-31]	CSR 32-bit offset for Format 7 Mode 16
324h	V_CSR_INQ_7_17	Mode_17	[0-31]	CSR 32-bit offset for Format 7 Mode 17
328h	V_CSR_INQ_7_18	Mode_18	[0-31]	CSR 32-bit offset for Format 7 Mode 18
32Ch	V_CSR_INQ_7_19	Mode_19	[0-31]	CSR 32-bit offset for Format 7 Mode 19
330h	V_CSR_INQ_7_20	Mode_20	[0-31]	CSR 32-bit offset for Format 7 Mode 20
334h	V_CSR_INQ_7_21	Mode_21	[0-31]	CSR 32-bit offset for Format 7 Mode 21

Offset	Name	Field	Bit	Description
338h	V_CSR_INQ_7_22	Mode_22	[0-31]	CSR 32-bit offset for Format 7 Mode 22
33Ch	V_CSR_INQ_7_23	Mode_23	[0-31]	CSR 32-bit offset for Format 7 Mode 23
340h	V_CSR_INQ_7_24	Mode_24	[0-31]	CSR 32-bit offset for Format 7 Mode 24
344h	V_CSR_INQ_7_25	Mode_25	[0-31]	CSR 32-bit offset for Format 7 Mode 25
348h	V_CSR_INQ_7_26	Mode_26	[0-31]	CSR 32-bit offset for Format 7 Mode 26
34Ch	V_CSR_INQ_7_27	Mode_27	[0-31]	CSR 32-bit offset for Format 7 Mode 27
350h	V_CSR_INQ_7_28	Mode_28	[0-31]	CSR 32-bit offset for Format 7 Mode 28
354h	V_CSR_INQ_7_29	Mode_29	[0-31]	CSR 32-bit offset for Format 7 Mode 29
358h	V_CSR_INQ_7_30	Mode_30	[0-31]	CSR 32-bit offset for Format 7 Mode 30
35Ch	V_CSR_INQ_7_31	Mode_31	[0-31]	CSR 32-bit offset for Format 7 Mode 31

D.8 General Camera Operation

The following settings control general status and monitoring of the camera:

- Memory Channel Registers (below)
- Device Information (on page 93)
- Camera Memory (on page 95)
- Firmware Information (on page 96)

D.8.1 Memory Channel Registers

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 values of the following registers are saved in memory channels.

Register Name	Offset
CURRENT_FRAME_RATE	600h
CURRENT_VIDEO_MODE	604h
CURRENT_VIDEO_FORMAT	608h
CAMERA_POWER	610h
CUR_SAVE_CH	620h
BRIGHTNESS	800h
AUTO_EXPOSURE	804h
SHARPNESS	808h
WHITE_BALANCE	80Ch
HUE	810h
SATURATION	814h
GAMMA	818h
SHUTTER	81Ch
GAIN	820h
IRIS	824h
FOCUS	828h
TRIGGER_MODE	830h
TRIGGER_DELAY	834h
FRAME_RATE	83Ch
PAN	884h
TILT	888h
ABS_VAL_AUTO_EXPOSURE	908h
ABS_VAL_SHUTTER	918h
ABS_VAL_GAIN	928h

Register Name	Offset
ABS_VAL_BRIGHTNESS	938h
ABS_VAL_GAMMA	948h
ABS_VAL_TRIGGER_DELAY	958h
ABS_VAL_FRAME_RATE	968h
IMAGE_DATA_FORMAT	1048h
AUTO_EXPOSURE_RANGE	1088h
AUTO_SHUTTER_RANGE	1098h
AUTO_GAIN_RANGE	10A0h
GPIO_XTRA	1104h
SHUTTER_DELAY	1108h
GPIO_STRPAT_CTRL	110Ch
GPIO_CTRL_PIN_x	1110h, 1120h, 1130h, 1140h
GPIO_XTRA_PIN_x	1114h, 1124h, 1134h, 1144h
GPIO_STRPAT_MASK_PIN_x	1118h, 1128h, 1138h, 1148h
FRAME_INFO	12F8h
IMAGE_POSITION	008h
IMAGE_SIZE	00Ch
COLOR_CODING_ID	010h
UDP_PORT	1F1Ch
DESTINATION_IP	1F34h

D.8.1.1 MEMORY_SAVE: 618h

All channels can be reset back to the original factory defaults by writing the value 0xDEAFBEEF to Memory_Save (ignores MEM_SAVE_CH).

Format:

Field	Bit	Description
Memory_Save	[0]	1 = Current status, modes, and values are saved to MEM_SAVE_CH (Self cleared)
	[1-31]	Reserved

D.8.1.2 MEM_SAVE_CH: 620h

Format:

Field	Bit	Description
Mem_Save_Ch	[0-3]	Specifies the write channel for Memory_Save command. Shall be >=0001 (0 is for factory default settings) See BASIC_FUNC_INQ register.
	[4-31]	Reserved

D.8.1.3 CUR_MEM_CH: 624h

Format:

Field	Bit	Description
Cur_Mem_Ch	[0-3]	Read: Reports the current memory channel number in use Write: Loads the camera status, modes and values from the specified memory channel
	[4-31]	Reserved

D.8.2 Device Information

Use the following to obtain information about the camera.

Pixel Clock Frequency—This specifies the current pixel clock frequency (in Hz) in IEEE-754 32-bit floating point format. The camera pixel clock defines an upper limit to the rate at which pixels can be read off the image sensor.

Horizontal Line Frequency—This specifies the current horizontal line frequency in Hz in IEEE-754 32-bit floating point format.

Serial Number—This specifies the unique serial number of the camera.

Main Board Information—This specifies the type of camera (according to the main printed circuit board).

Sensor Board Information—This specifies the type of imaging sensor used by the camera.

D.8.2.1 SERIAL_NUMBER: 1F20h

Format:

Field	Bit	Description
Serial_Number	[0-31]	Unique serial number of camera (read-only)

D.8.2.2 MAIN_BOARD_INFO: 1F24h

Format:

Field	Bit	Description
Major_Board_Design	[0-11]	0x6: Ladybug Head 0x7: Ladybug Base Unit 0x10: Flea 0x18: Dragonfly2 0x19: Flea2 0x1A: Firefly MV 0x1C: Bumblebee2 0x1F: Grasshopper 0x22: Grasshopper2 0x21: Flea2G-13S2 0x24: Flea2G-50S5 0x26: Chameleon 0x27: Grasshopper Express 0x29: Flea3 FireWire 14S3/20S4 0x2A: Flea3 FireWire 03S3 0x2B: Flea3 FireWire 03S1 0x2F: Flea3 GigE 14S3/20S4 0x32: Flea3 GigE 13S2 0x34: Flea3 USB 3.0 0x36: Zebra2 0x39: Flea3 GigE 03S2/08S2 0x3E: Flea3 GigE 50S5 0x3F: Flea3 GigE 28S4 0x40: Flea3 GigE 03S1
Minor_Board_Rev	[12-15]	Internal use
Reserved	[16-31]	Reserved

D.8.2.3 VOLTAGE: 1A50h – 1A54h

Format:

Offset	Name	Field	Bit	Description
1A50h	VOLTAGE_LO_INQ	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
		-	[1-7]	Reserved
			[8-19]	Number of voltage registers supported
		-	[20-31]	Reserved
1A54h	VOLTAGE_HI_INQ		[0-31]	32-bit offset of the voltage CSRs, which report the current voltage in Volts using the 32-bit floating-point IEEE/REAL*4 format.

D.8.2.4 CURRENT: 1A58h – 1A5Ch

Format:

Offset	Name	Field	Bit	Description
1A58h	CURRENT_LO_INQ	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
			[1-7]	Reserved
			[8-19]	Number of current registers supported
			[20-31]	Reserved
1A5Ch	CURRENT_HI_INQ		[0-31]	32-bit offset of the current registers, which report the current in amps using the 32-bit floating-point IEEE/REAL*4 format.

D.8.2.5 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.8.2.6 CAMERA_POWER: 610h

Format:

Field	Bit	Description
Cam_Pwr_Ctrl	[0]	Read: 0: Camera is powered down, or in the process of powering up (i.e., bit will be zero until camera completely powered up), 1: Camera is powered up Write: 0: Begin power-down process, 1: Begin power-up process
	[1-30]	Reserved
Camera_Power_Status	[31]	Read only Read: the pending value of Cam_Pwr_Ctrl

D.8.2.7 PIXEL_CLOCK_FREQ: 1AF0h

Format:

Field	Bit	Description
Pixel_Clock_Freq	[0-31]	Pixel clock frequency in Hz (read-only).

D.8.2.8 HORIZONTAL_LINE_FREQ: 1AF4h

Format:

Field	Bit	Description
Horizontal_Line_Freq	[0-31]	Horizontal line frequency in Hz (read-only).

D.8.3 Camera Memory

D.8.3.1 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.8.3.2 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.8.3.3 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.8.4 Firmware Information

D.8.4.1 FIRMWARE_VERSION: 1F60h

This register contains the version information for the currently loaded camera firmware.

Format:

Field	Bit	Description
Major	[0-7]	Major revision number
Minor	[8-15]	Minor revision number
Type	[16-19]	Type of release: 0: Alpha 1: Beta 2: Release Candidate 3: Release
Revision	[20-31]	Revision number

D.8.4.2 FIRMWARE_BUILD_DATE: 1F64h

Format:

Field	Bit	Description
Build_Date	[0-31]	Date the current firmware was built in Unix time format (read-only)

D.8.4.3 FIRMWARE_DESCRIPTION: 1F68-1F7Ch

Null padded, big-endian string describing the currently loaded version of firmware.

D.9 Input/Output Control

The following settings are used for input/output control:

- GPIO Pin Control (below)
- GPIO Xtra Control (for Pulse Width Modulation) (on the next page)
- GPIO Strobe Control Registers (on page 100)
- Strobe Output Registers (on page 100)
- Serial Communication Registers (on page 101)

D.9.1 GPIO_CTRL_PIN: 1110h-1140h

These registers provide control over the GPIO pins.

Pin	Register	
0	GPIO_CTRL_PIN_0	1110h
1	GPIO_CTRL_PIN_1	1120h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-11]	Reserved
Pin_Mode	[12-15]	Current GPIO Mode: 0: Input 1: Output 2: Asynchronous Trigger 3: Strobe 4: Pulse width modulation (PWM)
	[16-30]	For Modes 0, 1, and 2: Reserved For Mode 4 (PWM:) see below
Data	[31]	For Modes 0, 1, and 2: Data field 0 = 0 V (falling edge), 1 = +3.3 V (rising edge) For Mode 4 (PWM): see below
Pwm_Count	[16-23]	Number of PWM pulses Read: The current count; counts down the remaining pulses. After reaching zero, the count does not automatically reset to the previously-written value. Write: Writing the number of pulses starts the PWM. Write 0xFF for infinite pulses. (Requires write of 0x00 before writing a different value.)
	[24]	Reserved
En_Pin	[25-27]	The GPIO pin to be used as a PWM enable i.e. the PWM continues as long as the En_Pin is held in a certain state (high or low).
	[28]	Reserved

Field	Bit	Description
Disable_Pol	[29]	Polarity of the PWM enable pin (En_Pin) that will disable the PWM. If this bit is 0, the PWM is disabled when the PWM enable pin goes low.
En_En	[30]	0: Disable enable pin (En_Pin) functionality 1: Enable En_Pin functionality
Pwm_Pol	[31]	Polarity of the PWM signal 0: Low, 1: High

D.9.2 GPIO_XTRA_PIN: 1114h-1144h

These registers contain mode specific data for the GPIO pins. Units are ticks of a 1.024MHz clock.

Pin	Register	
0	GPIO_XTRA_PIN_0	1114h
1	GPIO_XTRA_PIN_1	1124h

Format:

Field	Bit	Description
Mode_Specific_1	[0-15]	GPIO_MODE_4: Low period of PWM pulse (if Pwm_Pol = 0)
Mode_Specific_2	[16-31]	GPIO_MODE_4: High period of PWM pulse (if Pwm_Pol = 0)

D.9.3 GPIO_CONTROL: 1100h

This register provides status information about the camera's GPIO pins.

Opto-isolated input pins with pull-up resistors report a value of '1' when unconnected. Consult your camera's Technical Reference manual for GPIO pinout details.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-11]	Reserved
Pin_Count	[12-15]	Number of available GPIO pins
	[16-27]	Reserved
Value_3	28]	Value of IO3 0: Voltage low; 1: Voltage high
Value_2	[29]	Value of IO2 0: Voltage low; 1: Voltage high
Value_1	[30]	Value of IO1 0: Voltage low; 1: Voltage high
Value_0	[31]	Value of IO0 0: Voltage low; 1: Voltage high

D.9.4 GPIO_STRPAT_CTRL: 110Ch

This register provides control over a shared 4-bit counter with programmable period. When the *Current_ Count* equals N a GPIO pin will only output a strobe pulse if bit[N] of the GPIO_STRPAT_MASK_PIN_x register's *Enable_Pin* field is set to '1'.

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-18]	Reserved
Count_Period	[19-23]	Controls the period of the strobe pattern Valid values: 1..16
	[24-27]	Reserved
Current_Count	[28-31]	Read-only The value of the bit index defined in GPIO_x_STRPAT_MASK that will be used during the next image's strobe. *Current_Count* increments at the same time as the strobe start signal occurs.

D.9.5 GPIO_STRPAT_MASK_PIN: 1118h-1148h

These registers define the actual strobe pattern to be implemented by GPIO pins in conjunction with the *Count_Period* defined in GPIO_STRPAT_CTRL register 110Ch.

For example, if *Count_Period* is set to '3', bits 16-18 of the *Enable_ Mask* can be used to define a strobe pattern. An example strobe pattern might be bit 16=0, bit 17=0, and bit 18=1, which will cause a strobe to occur every three frames (when the *Current_Count* is equal to 2).

Pin	Register	
0	GPIO_STRPAT_MASK_PIN_0	1118h
1	GPIO_STRPAT_MASK_PIN_1	1128h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-15]	Reserved
Enable_Mask	[16-31]	Bit field representing the strobe pattern used in conjunction with *Count_Period* in GPIO_STRPAT_CTRL 0: Do not output a strobe, 1: Output a strobe

D.9.6 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.

Format:

Field	Bit	Description
Strobe_Start	[0]	Current Mode 0: Strobe start is relative to start of integration (default) 1: Strobe start is relative to external trigger
	[1-31]	Reserved

D.9.7 Serial Input/Output Registers

This section describes the control and inquiry registers for the serial input/output (SIO) control functionality.

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)*

Offset	Name	Field	Bit	Description
488h	SIO_CONTROL_CSR_INQ	SIO_Control_Quadlet_Offset	[0-31]	32-bit offset of the SIO CSRs from the base address of initial register space
Base + 0h	SERIAL_MODE_REG	Baud_Rate	[0-7]	*Baud rate setting* Read: Get current baud rate Write: Set baud rate 0: 300 bps 1: 600 bps 2: 1200 bps 3: 2400 bps 4: 4800 bps 5: 9600 bps 6: 19200 bps 7: 38400 bps 8: 57600 bps 9: 115200 bps 10: 230400 bps Other values reserved
		Char_Length	[8-15]	Character length setting Read: Get data length Write: Set data length (must not be 0) 7: 7 bits, 8: 8 bits Other values reserved

Offset	Name	Field	Bit	Description
		Parity	[16-17]	*Parity setting* Read: Get current parity Write: Set parity 0: None, 1: Odd, 2: Even
		Stop_Bit	[18-19]	*Stop bits* Read: Get current stop bit Write: Set stop bit 0: 1, 1: 1.5, 2: 2
			[20-23]	Reserved
		Buffer_ Size_Inq	[24-31]	*Buffer Size (Read-Only)* This field indicates the maximum size of the receive/transmit data buffer. See also SIO Buffers on page 26 If this value=1, *Buffer_Status_Control* and *SIO_Data_Register* characters 1-3 should be ignored.
Base + 4h	SERIAL_ CONTROL_REG	RE	[0]	Receive enable Indicates if the camera's ability to receive data has been enabled. Enabling this register causes the receive capability to be immediately started. Disabling this register causes the data in the buffer to be flushed. Read: Current status Write: 0 Disable, 1: Enable
		TE	[1]	*Transmit enable* Indicates if the camera's ability to transmit data has been enabled. Enabling this register causes the transmit capability to be immediately started. Disabling this register causes data transmission to stop immediately, and any pending data is discarded. Read: Current status Write: 0: Disable, 1: Enable
		-	[2-7]	Reserved
	SERIAL_ STATUS_REG	TDRD	[8]	*Transmit data buffer ready (read only)* Indicates if the transmit buffer is ready to receive data from the user. It will be in the Ready state as long as *TBUF_ST* != 0 and *TE* is enabled. Read only 0: Not ready, 1: Ready
		-	[9]	Reserved

Offset	Name	Field	Bit	Description
		RDRD	[10]	*Receive data buffer ready (read only)* Indicates if the receive buffer is ready to be read by the user. It will be in the Ready state as long as *RBUF_ST* != 0 and *RE* is enabled. Read only 0: Not ready, 1: Ready
		-	[11]	Reserved
		ORER	[12]	*Receive buffer over run error* Read: Current status Write: 0: Clear flag, 1: Ignored
		FER	[13]	*Receive data framing error* Read: Current status Write: 0: Clear flag, 1: Ignored
		PER	[14]	*Receive data parity error* Read: Current status Write: 0: Clear flag, 1: Ignored
		-	[15-31]	Reserved
Base + 8h	RECEIVE_ BUFFER_ STATUS_ CONTROL	RBUF_ST	[0-8]	*SIO receive buffer status* Indicates the number of bytes that have arrived at the camera but have yet to be queued to be read. Read: Valid data size of current receive buffer Write: Ignored
		RBUF_CNT	[8-15]	*SIO receive buffer control* Indicates the number of bytes that are ready to be read. Read: Remaining data size for read Write: Set input data size
		-	[16-31]	Reserved
Base + Ch	TRANSMIT_ BUFFER_ STATUS_ CONTROL	TBUF_ST	[0-8]	*SIO output buffer status* Indicates the minimum number of free bytes available to be filled in the transmit buffer. It will count down as bytes are written to any of the SIO_ DATA_REGISTERs starting at 2100h. It will count up as bytes are actually transmitted after a write to *TBUF_CNT*. Although its maximum value is 255, the actual amount of available buffer space may be larger. Read: Available data space of transmit buffer Write: Ignored

Offset	Name	Field	Bit	Description
		TBUF_CNT	[8-15]	*SIO output buffer control* Indicates the number of bytes that have been stored since it was last written to. Writing any value to *TBUF_CNT* will cause it to go to 0. Writing a number less than its value will cause that many bytes to be transmitted and the rest thrown away. Writing a number greater than its value will cause that many bytes to be written - its value being valid and the remainder being padding. Read: Written data size to buffer Write: Set output data size for transmit.
		-	[16-31]	Reserved
Base + 100h	SIO_DATA_ REGISTER	Char_0	[0-7]	*Character_0* Read: Read character from receive buffer. Padding data if data is not available. Write: Write character to transmit buffer. Padding data if data is invalid.
		Char_1	[8-16]	*Character_1* Read: Read character from receive buffer+1. Padding data if data is not available. Write: Write character to transmit buffer+1. Padding data if data is invalid.
		Char_2	[17-23]	*Character_2* Read: Read character from receive buffer+2. Padding data if data is not available. Write: Write character to transmit buffer+2. Padding data if data is invalid.
		Char_3	[24-31]	*Character_3* Read: Read character from receive buffer+3. Padding data if data is not available. Write: Write character to transmit buffer+3. Padding data if data is invalid.
Base + 104h : Base + 1FFh	SIO_DATA_ REGISTER_ALIAS		[0-31]	Alias SIO_Data_Register area for block transfer.

D.10 Video Mode Control and Status Registers

These registers provide partial image size format (Format 7, Mode x) information.

D.10.1 FRAME_RATE: 83Ch

Formulas for converting the fixed point (relative) values to floating point (absolute) values are not provided. Users wishing to work with real-world values should refer to Absolute Value CSRs (page 78).

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control in the Value field, 1: Control in the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
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
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic
	[8-19]	Reserved
Value	[20-31]	Value. A write to this value in ‘Auto’ mode will be ignored.

D.10.2 CURRENT_FRAME_RATE: 600h

Format:

Field	Bit	Description
Cur_V_Frm_Rate	[0-2]	Current frame rate FrameRate_0 .. FrameRate_7
	[3-31]	Reserved.

D.10.3 CURRENT_VIDEO_MODE: 604h

Format:

Field	Bit	Description
Cur_V_Mode	[0-3]	Current video mode Mode_0 .. Mode_8
	[4-31]	Reserved.

D.10.4 CURRENT_VIDEO_FORMAT: 608h

Format:

Field	Bit	Description
Cur_V_Format	[0-2]	Current video format Format_0 .. Format_7
	[3-31]	Reserved.

D.10.5 FORMAT_7_RESIZE_INQ: 1AC8h

This register reports all internal camera processes being used to generate images in the current video mode. For example, users can read this register to determine if pixel binning and/or subsampling is being used to achieve a non-standard custom image size.

This register is read-only.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-7]	Reserved
Num_Cols	[8-11]	Number of columns being binned/subsampled, minus 1 (e.g., if combining 4 columns together, this register will report a value of 3)
Num_Rows	[12-15]	Number of rows binned/subsampled, minus 1 (e.g., if combining 4 columns together, this register will report a value of 3)
	[16-23]	Reserved
V_Pre_Color	[24]	Vertical subsampling/downsampling performed before color processing 0: Off, 1: On
H_Pre_Color	[25]	Horizontal subsampling/downsampling performed before color processing 0: Off, 1: On
V_Post_Color	[26]	Vertical subsampling/downsampling performed after color processing 0: Off, 1: On
H_Post_Color	[27]	Horizontal subsampling/downsampling performed after color processing 0: Off, 1: On
V_Bin	[28]	Standard vertical binning (addition of adjacent lines within horizontal shift register) 0: Off, 1: On

Field	Bit	Description
H_Bin	[29]	Standard horizontal binning (addition of adjacent lines within horizontal shift register) 0: Off, 1: On
V_Bayer_Bin	[30]	Vertical bayer binning (addition of adjacent even/odd lines within the interline transfer buffer) 0: Off, 1: On
H_Bayer_Bin	[31]	Horizontal bayer binning (addition of adjacent even/odd columns within the horizontal shift register) 0: Off, 1: On

D.10.6 Inquiry Registers for Custom Video Mode Offset Addresses

The following set of registers indicates the locations of the custom video mode base registers. These offsets are relative to the base offset 0xFFFF F0F0 0000.

Table D.2: Custom Video Mode Inquiry Register Offset Addresses

Offset	Name	Field	Bit	Description
2E0h	V_CSR_INQ_7_0	Mode_0	[0-31]	32-bit offset for Mode 0
2E4h	V_CSR_INQ_7_1	Mode_1	[0-31]	32-bit offset for Mode 1
2E8h	V_CSR_INQ_7_2	Mode_2	[0-31]	32-bit offset for Mode 2
2ECh	V_CSR_INQ_7_3	Mode_3	[0-31]	32-bit offset for Mode 3
2F0h	V_CSR_INQ_7_4	Mode_4	[0-31]	32-bit offset for Mode 4
2F4h	V_CSR_INQ_7_5	Mode_5	[0-31]	32-bit offset for Mode 5
2F8h	V_CSR_INQ_7_6	Mode_6	[0-31]	32-bit offset for Mode 6
2FCh	V_CSR_INQ_7_7	Mode_7	[0-31]	32-bit offset for Mode 7
300h	V_CSR_INQ_7_8	Mode_8	[0-31]	32-bit offset for Mode 8

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)*

D.10.6.1 Image Size and Position

These registers are inquiry registers for maximum image size and unit size, and to determine an area of required data.

Format:

Address	Name	Field	Bit	Description
Base + 000h	MAX_IMAGE_SIZE_INQ	Hmax	[0-15]	Maximum horizontal pixel number Hmax = Hunit * n = Hposunit*n3 (n, n3 are integers)
		Vmax	[16-31]	Maximum vertical pixel number Vmax = Vunit * m = Vposunit*m3 (m, m3 are integers)
Base + 004h	UNIT_SIZE_INQ	Hunit	[0-15]	Horizontal unit pixel number
		Vunit	[16-31]	Vertical unit pixel number

Address	Name	Field	Bit	Description
Base + 04Ch	UNIT_POSITION_INQ	Hposunit	[0-15]	Horizontal unit pixel number for position If read value of Hposunit is 0, Hposunit = Hunit for IIDC 1.20 compatibility.
		Vposunit	[16-31]	Vertical unit number for position If read value of Vposunit is 0, Vposunit = Vunit for IIDC 1.20 compatibility.
Base + 008h	IMAGE_POSITION	Left	[0-15]	Left position of requested image region (pixels) Left = Hposunit * n1 Left + Width <= Hmax
		Top	[16-31]	Top position of requested image region (pixels) Top = Vposunit * m1 Top + Height <= Vmax
Base + 00Ch	IMAGE_SIZE	Width	[0-15]	Width of requested image region (pixels) Width = Hunit * n2
		Height	[16-31]	Height of requested image region (pixels) Height = Vunit * m2 (n1, n2, m1, m2 are integers)

D.10.6.2 COLOR_CODING_ID and COLOR_CODING_INQ

The COLOR_CODING_INQ register describes the color-coding capability of the system. Each coding scheme has its own ID number. The required color-coding scheme must be set to COLOR_CODING_ID register as the ID number.

Format:

Address	Name	Field	Bit	Description	ID
Base + 010h	COLOR_CODING_ID	Coding_ID	[0-7]	Color coding ID from COLOR_CODING_INQ register	N/A
			[8-31]	Reserved	N/A
Base + 014h	COLOR_CODING_INQ	Mono8	[0]	Y only. Y=8bits, non compressed	0
		4:1:1 YUV8	[1]	4:1:1, Y=U=V= 8bits, non compressed	1
		4:2:2 YUV8	[2]	4:2:2, Y=U=V=8bits, non compressed	2
		4:4:4 YUV8	[3]	4:4:4, Y=U=V=8bits, non compressed	3
		RGB8	[4]	R=G=B=8bits, non compressed	4
		Mono16	[5]	Y only, Y=16bits, non compressed	5
		RGB16	[6]	R=G=B=16bits, non compressed	6
		Signed Mono16	[7]	Y only, Y=16 bits, non compressed (signed integer)	7
		Signed RGB16	[8]	R=G=B=16 bits, non compressed (signed integer)	8
		Raw8	[9]	Raw data output of color filter sensor, 8 bits	9
		Raw16	[10]	Raw data output of color filter sensor, 16 bits	10
		Mono12	[11]	Y only. Y=12 bits, non compressed	
		Raw12	[12]	Raw data output of color filter sensor, 12 bits	
			[13-31]	Reserved	11-31

D.10.6.3 FRAME_INTERVAL_INQ

Format:

Address	Name	Field	Bit	Description
Base + 050h	FRAME_INTERVAL_ INQ	FrameInterval	[0-31]	Current frame interval (seconds) based on the current camera conditions, including exposure time. The reciprocal value of this (1 / FrameInterval) is the frame rate of the camera. IEEE/REAL*4 floating-point value (see *Determining Absolute Value Register Values* (page 78)) If 0, the camera can't report the value and it should be ignored.

D.10.7 IMAGE_DATA_FORMAT: 1048h (IIDC 1.31)

This register allows the user to specify various image data format parameters.

Mirror_Image_Ctrl allows the user to toggle between normal and mirror (horizontally flipped) image modes.

Bayer_Mono_Ctrl allows the user to control whether non-Format 7 Y8 or Y16 monochrome modes on a color camera will output monochrome (greyscale) or raw Bayer data.

Selecting a half-width, half-height image size and monochrome pixel format, such as 800 x 600 Y8, using non-Format 7 modes provides a monochrome binned image. In some cases, enabling raw Bayer output in mono mode provides a raw Bayer region of interest of 800 x 600, centered within the larger pixel array. This has an effect on the field of view.

Y16_Data_Format controls the endianness of Y16 images – either IIDC 1394 DCAM-compliant mode (default) or PGR-specific (Intel-compatible) mode – as described below.

IIDC 1394 DCAM Y16 Mode			PGR-specific Y16 Mode		
Description	Data Format		Description	Data Format	
Actual bit depth: Dependent on ADC Bit alignment: MSB Byte alignment: Big Endian	0-7	8-15	Actual bit depth: Dependent on ADC Bit alignment: MSB Byte alignment: Little Endian	0-7	8-15
	98765432	10xxxxxx		10xxxxxx	98765432

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
Reserved	[1-22]	Reserved
Mirror_Image_Ctrl	[23]	Control horizontally flipped image modes 0: Disable image flip 1: Enable image flip
Bayer_Mono_Ctrl	[24]	Control raw Bayer output in non-Format 7 mono modes 0: Disable 1: Enable

Field	Bit	Description
Reserved	[25-30]	Reserved
Y16_Data_Format	[24-31]	Value: 0: PGR-specific mode 1: DCAM-compliant mode (default)

D.11 Asynchronous Trigger Settings

For information about working with the trigger registers in your FlyCapture application, refer to the AsyncTriggerEx example program, available with the FlyCapture SDK.

Trigger Mode—This controls the trigger mode. When trigger mode is enabled, frame rate is changed from Auto to Off state. This change affects the maximum shutter time (page 43). If trigger mode is disabled, frame rate remains in the Off state.

Trigger Delay—This provides control over the time delay, depending on the current mode:

- In Asynchronous trigger mode: controls the delay between the trigger event and the start of integration (shutter open).

Software Trigger—This allows the user to generate a software asynchronous trigger.

D.11.1 TRIGGER_MODE: 830h

Control of the register is via the *ON_OFF* bit and the *Trigger_Mode* and *Parameter* fields.

Format

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the Value field, 1: Control with the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.
	[2-5]	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
Trigger_Polarity	[7]	Select trigger polarity (except for Software_Trigger) 0: Trigger active low, 1: Trigger active high
Trigger_Source	[8-10]	Select trigger source: used to select which GPIO pin will be used for external trigger purposes. Sets trigger source ID from *Trigger_Source_Inq* field of TRIGGER_INQ register(page 83).
Trigger_Value	[11]	Trigger input raw signal value: used to determine the current raw signal value on the pin. Read only 0: Low, 1: High
	[8-11]	Reserved
Trigger_Mode	[12-15]	Trigger mode (Trigger_Mode_0..15): used to set the trigger mode to be used. For more information, see Asynchronous Triggering. Query the *Trigger_Mode_Inq* fields of the TRIGGER_INQ register for available trigger modes.
	[16-19]	Reserved
Parameter	[20-31]	Parameter for trigger function, if required (optional)

D.11.2 TRIGGER_DELAY: 834h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the Value field, 1: Control with the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.
	[2-5]	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
	[7-19]	Reserved
Value	[20-31]	Value.

D.11.3 SOFTWARE_TRIGGER: 62Ch

Bit 0 of this register indicates if the camera is ready to be triggered again for both software and hardware triggering.

Format:

Field	Bit	Description
Software_Trigger	[0]	This bit automatically resets to zero in all trigger modes except Trigger Mode 3. Read: 0: Ready, 1: Busy Write: 0: Reset software trigger, 1: Set software trigger

D.12 Controlling Imaging Parameters

The registers in this section are used to control imaging parameters for the camera.

D.12.1 Imaging Parameters: 800h-888h

The following imaging parameters share the same register format.

Parameter	Register
Brightness	800h
Sharpness	808h
Hue	810h
Saturation	814h
Gamma	818h
Gain	820h
Iris	824h
Focus	828h
Pan	884h
Tilt	888h

These imaging parameters are defined by **modes** and **values**.

There are three modes:

Mode	Description
On/Off	Determines if the feature is on. If off, values are fixed and not controllable.
Auto/Manual	If the feature is on, determines if the feature is in automatic or manual mode. If manual, values can be set.
One Push	If the feature is in manual mode, the camera executes once automatically and then returns to manual mode.

The value field in this register can be set in three ways:

Method	Description
Absolute	The user sets the value is set via the absolute register. The *Value* field becomes read only and reflects the converted absolute value.
Manual	The user sets the value in the *Value* field. The absolute register becomes read only and contains the current value.
Automatic	The value is set automatically by another register and both the *Value* field and the absolute register become read only.

Formulas for converting the fixed point (relative) values to floating point (absolute) values are not provided. Users wishing to work with real-world values should refer to Absolute Value CSRs *(page 78)**.*

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control in the Value field, 1: Control in the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
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
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic
	[8-19]	Reserved
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

D.12.2 LUT: 1A40h – 1A44h (IIDC 1.31)

Cameras using the IIDC Specification version 1.31 must use the following lookup table registers.

This register allows the user to access and control a lookup table (LUT), with entries stored onboard the camera. Changes to GAMMA are translated to writes of the LUT CSR registers. The LUT will also be modifed under the following circumstances:

- Camera reinitialization via the INITIALIZE register 000h
- Changing the CURRENT_VIDEO_MODE or CURRENT_VIDEO_FORMAT registers 604h or 608h
- Changing the GAMMA register 818h or ABS_VAL_GAMMA register
- Changing the WHITE_BALANCE register 80Ch (SCOR-13FF only)
- Writing the AUTO_EXPOSURE_RANGE register 108Ch (Flea only)

Offset	Name	Field	Bit	Description
1A40h	LUT_LO_CTRL	Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
			[1-2]	Reserved
		Num_Channels	[3-5]	Number of channels
		ON_OFF	[6]	Write: ON or OFF for this feature Read: Read a status 0: OFF, 1: ON If this bit = 0, other fields are read only
			[7]	Reserved
		Bit_Depth	[8-15]	Bit depth of the lookup table
		Entries	[16-31]	Number of entries in the table
1A44h	LUT_HI_INQ		[0-31]	32-bit offset of the lookup table

D.12.3 WHITE_BALANCE: 80Ch

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the Value field, 1: Control with the Absolute Value CSR If this bit is 1, then Value is ignored
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
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
A_M_Mode	[7]	Read: read the current mode. Write: Set the mode. 0: Manual, 1: Auto
U_Value/B_Value	[8-19]	Blue Value. A write to this value in ‘Auto’ mode will be ignored.
V_Value/R_Value	[20-31]	Red Value. A write to this value in ‘Auto’ mode will be ignored.

D.12.4 BAYER_TILE_MAPPING: 1040h

This 32-bit read only register specifies the sense of the cameras' Bayer tiling. Various colors are indicated by the ASCII representation of the first letter of their name.

Color	ASCII
Red (R)	52h
Green (G)	47h
Blue (B)	42h
Monochrome (Y)	59h

For example, 0x52474742 is RGGB and 0x59595959 is YYYY.

Format

Field	Bit	Description
Bayer_Sense_A	[0-7]	ASCII representation of the first letter of the color of pixel (0,0) in the Bayer tile.
Bayer_Sense_B	[8-15]	ASCII representation of the first letter of the color of pixel (0,1) in the Bayer tile.
Bayer_Sense _C	[16-24]	ASCII representation of the first letter of the color of pixel (1,0) in the Bayer tile.
Bayer_Sense _D	[25-31]	ASCII representation of the first letter of the color of pixel (1,1) in the Bayer tile.

D.12.5 SHUTTER: 81Ch

This register has three states:

State	Description
Manual/Abs	The shutter value is set by the user via the ABS_VAL_SHUTTER register (page 78). The *Value* field becomes read only and reflects the converted value of the ABS_VAL_SHUTTER register.
Manual	The user sets the shutter value via the *Value* field. The ABS_VAL_SHUTTER register becomes read only and contains the current shutter time.
Auto	The shutter value is set by the auto exposure controller (if enabled) (page 44). Both the *Value* field and the ABS__VAL_SHUTTER register become read only.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the *Value* field, 1: Control with the Absolute value CSR. If this bit = 1, the value in the *Value* field is ignored.
	[2-4]	Reserved

Field	Bit	Description
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
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
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic
High_Value	[8-19]	Upper 4 bits of the shutter value available only in extended shutter mode (outside of specification).
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

D.12.6 AUTO_EXPOSURE: 804h

Formulas for converting the fixed point (relative) values to floating point (absolute) values are not provided. Users wishing to work with real-world values should refer to Absolute Value CSRs *(page 78)**.*

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the *Value* field, 1: Control with the Absolute value CSR. If this bit = 1, the value in the *Value* field is ignored.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
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

Field	Bit	Description
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic
High_Value	[8-19]	Upper 4 bits of the shutter value available only in extended shutter mode (outside of specification).
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

D.12.6.1 AUTO_EXPOSURE_RANGE: 1088h

Fixed point (relative) values must be specified. Do not specify absolute values.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-7]	Reserved
Min_Value	[8-19]	Lower bound
Max_Value	[20-31]	Upper bound

D.12.6.2 AUTO_SHUTTER_RANGE: 1098h

Fixed point (relative) values must be specified. Do not specify absolute values.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-5]	Reserved
Min_Dark_Noise	[6]	Minimizes dark current noise with extended shutter times. This feature is currently experimental. 0: Disable dark noise minimization, 1: Enable dark noise minimization
	[7]	Reserved
Min_Value	[8-19]	Lower bound
Max_Value	[20-31]	Upper bound

The actual range used is further restricted to match the current grab mode (see SHUTTER: 81Ch for the list of ranges).

D.12.6.3 AUTO_GAIN_RANGE: 10A0h

Format:

Field	Bit	Description
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]	Reserved
Min_Value	[8-19]	Lower bound
Max_Value	[20-31]	Upper bound

D.12.6.4 AE_ROI: 1A70 – 1A74h

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
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.12.7 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	
Insert_Info	[22]	Display image-specific information 0: Off 1: On	Region of Interest (ROI) position (See page 48)
	[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 48)

D.13 Troubleshooting

The following registers help with troubleshooting issues with the camera:

- Camera Diagnostics (on page 53)
- Pixel Defect Correction (on the next page)

D.13.1 Camera Diagnostics

There are a number of control and status registers that can be used for camera diagnostics.

D.13.1.1 INITIALIZE: 000h

Format:

Offset	Name	Field	Bit	Description
000h	INITIALIZE	Initialize	[0]	If this bit is set to 1, the camera will reset to its initial state and default settings. This bit is self-cleared.
			[1-31]	Reserved

D.13.1.2 TIME_FROM_INITIALIZE: 12E0h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Time_From_Init	[1-31]	Time in seconds since the camera was initialized.

D.13.1.3 XMIT_FAILURE: 12FCh

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Frame_Count	[1-31]	Read: Count of failed frame transmissions. Write: Reset.

D.13.1.4 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.13.1.5 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.13.1.6 TEST_PATTERN: 104Ch

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-30]	Reserved
Test_Pattern_1	[31]	Value 0: Disable test pattern, 1: Enable test pattern

D.13.2 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

Revision History

Revision	Date	Notes
1.1	October 31, 2008	Section 3.1: USB Connector: Added a reference to Knowledge Base Article 309: Using USB 2.0 PCI host adapter cards with USB cameras. Section 3.4: GPIO: Clarified voltage limit of VEXT pin.
1.2	January 13, 2009	Section 3.4: GPIO. Revised Table 1 to indicate that pins 5 and 6 are supported for input/output. Section 3.2: Cables. Added a link to usb.org website for additional information about cable lengths. Section 3.2: Cables. Added a link to usb.org website for additional information about cable lengths.
1.3	June 9, 2009	Added Region of Interest (ROI) frame rates to Table 4 in Section 4.4: Customizable Data Formats and Modes. Section 4.3.1: Clarified that Point Grey does not support the use of multiple USB 2.0 cameras streaming simultaneously on the same computer. Section 4.3.2:-Removed this section, which discusses maximum frame rate calculations for cameras on the same bus. Glosary: Fixed error in 1394b definition. Clarified in Section 1.3: Specifications that the camera can synchronize to an external hardware trigger or a software trigger. Clarified that the Chameleon complies with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules.
1.4	December 7, 2009	Clarified camera dimensions in Section 1.3 Camera Specifications. Added link to Knowledge Base Article 295 in Section 3.3: Camera Power. Section 4.4 Customizable Data Formats and Modes: Removed Mono8/16 formats from Format 7 Mode 0. These formats are no longer supported. Section 4.4.1 Calculating Format 7 Frame Rates: Updated equation.
1.5	March 17, 2011	Added information about M12 microlens holder for board-level camera models to section 2.2 Camera Dimensions and section 2.3 Lens Setup and Compatibility. Section 4.5.7 Pixel Binning and Region of Interest Modes Added binning information about Format 7 Mode 1 and Mode 2. Section 4.2 Standard Data Formats, Modes and Frame Rates Clarified that 30 FPS is achieved through pixel binning, with no color output. Section 1.8.1 Heat Dissipation Added information about location of temperature sensor. Added Section 1.9 Common CCD Artifacts Section 4.5.5 Extended Shutter Times: Added extended shutter times of camera in full resolution.

Revision	Date	Notes
1.6	July 12, 2011	Section 4.4 Customizable Data Formats and Modes: Clarified that when outputting in Raw8 or Raw16 format, the camera outputs color data only in 1280x960 resolution. In lower resolutions, the camera performs pixel binning, which destroys the Bayer tile pattern. Section 4.5.9.3 Minimum Trigger Pulse Length: Clarified the role of the signal debouncer. Section 4.4 Customizable Data Formats and Modes: Revised bytes per packet for Format 7 modes. Added Section 4.7.3 Flash Memory.
2.0	December 10, 2013	Reorganization of document Added Section 4.7.2 Serial Communication Using GPIO. Clarification of binning in Video Modes