Grasshopper Express

FireWire 1394b Digital Camera

Technical Reference Manual

Version 3.1

Revised 10/4/2012

FireWire

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.

Hardware Warranty

Point Grey Research®, Inc. (Point Grey) warrants to the Original Purchaser that the Camera Module provided with this package is guaranteed to be free from material and manufacturing defects for a period of Three years. Should a unit fail during this period, Point Grey will, at its option, repair or replace the damaged unit. Repaired or replaced units will be covered for the remainder of the original equipment warranty period. This warranty does not apply to units that, after being examined by Point Grey, have been found to have failed due to customer abuse, mishandling, alteration, improper installation or negligence. If the original camera module is housed within a case, removing the case for any purpose other than to remove the protective glass or filter over the sensor voids this warranty. This warranty does not apply to damage to any part of the optical path resulting from removal or replacement of the protective glass or filter over the camera, such as scratched glass or sensor damage.

Point Grey Research, Inc. expressly disclaims and excludes all other warranties, express, implied and statutory, including, but without limitation, warranty of merchantability and fitness for a particular application or purpose. In no event shall Point Grey Research, Inc. be liable to the Original Purchaser or any third party for direct, indirect, incidental, consequential, special or accidental damages, including without limitation damages for business interruption, loss of profits, revenue, data or bodily injury or death.

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, 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

List of Tables

List of Figures

About This Manual

This manual provides the user with a detailed specification of the Grasshopper Express 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 on page 156.)

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 on page 34.

Where to Find Information

Chapter	What You Will Find
1. Welcome	General camera specifications and specific model specifications (page 1) Imaging Performance specifications and Quantum Efficiency graphs Camera properties, including diagrams (page 9)
2. Getting Started	Preparation for installing the camera (page 18) Installation instructions (page 20) Introduction to camera controls (page 21)
3. General Operation	Powering the camera (page 24) Device Information (page 25) User Configuration sets (page 28) On-camera frame buffer (page 30) Flash memory (page 33) Firmware (page 34)
4. Input/Output Control	GPIO Modes (page 37) Programmable Strobe Output (page 38) Pulse Width Modulation (page 48) Serial Communication (page 49) GPIO Electrical Characteristics (page 57)
5. Video Formats, Modes, and Frame Rates	Overview and descriptions of Video Modes (page 64) Supported Formats, Modes, and Frame Rates for each model (page 67) Video Format and Mode CSRs (page 72)
6. Image Acquisition and Transmission	Isochronous Data Transfer (page 74) Asynchronous Triggering (page 77) and Supported Trigger Modes (page 79)

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 FlyCapture2 SDK.

Related Resources

Title	Link
Title of the resource	Link to the resource

1 Introduction

1.1 Welcome to Grasshopper Express

The Grasshopper Express camera series features a hardware platform designed to accommodate a wide range of high speed image sensors. The camera is housed in the same compact metal case as the original Grasshopper, and features enhanced opto-isolated GPIO, new hardware trigger modes, and improved imaging performance.

Related Knowledge Base Articles

Title	Article
Key features and benefits of the IEEE-1394b standard	Knowledge Base Article 206

1.2 Grasshopper Express Specifications

MODEL	VERSION	MP	IMAGING SENSOR
GX-FW-10K3M-C	Mono	1.0 MP	▪ Kodak KAI-01050 CCD 1/2", 1/2", 5.5 µm ▪ Global Shutter ▪ 1024x1024 at 70 FPS
GX-FW-28S5C-C **GX-FW-28S5M-C**	Color Mono	2.8 MP	▪ Sony ICS674 CCD 2/3", 2/3", 4.54 µm ▪ Global Shutter ▪ 1932x1452 at 26 FPS
GX-FW-60S6C-C **GX-FW-60S6M-C**	Color Mono	6.0 MP	▪ Sony ICX694 CCD, 1", 4.54 µm ▪ Global Shutter ▪ 2736x2192 at 11 FPS

	All Grasshopper Express Models
A/D Converter	14-bit
Video Data Output	8, 12, 16 and 24-bit digital data
Image Data Formats	Y8, Y16, Mono8, Mono12, Mono16 (all models) RGB, YUV411, YUV422, YUV444, Raw8, Raw12, Raw16 (color models)
Partial Image Modes	Pixel binning and region of interest (ROI) modes
Image Processing	Gamma, lookup table, hue, saturation, and sharpness
Gain	Automatic/Manual/One-Push Gain modes
	0 dB to 24 dB
Gamma	0.50 to 4.00
White Balance	Automatic/manual modes, programmable via software
High Dynamic Range	Cycle 4 gain and exposure presets

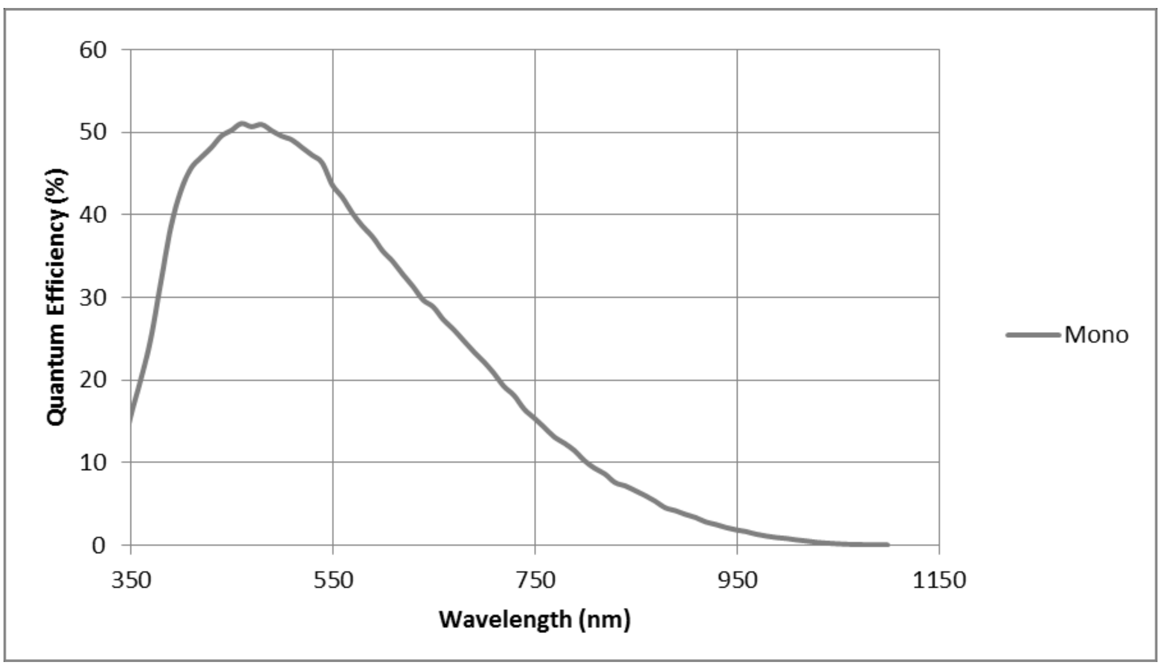
	All Grasshopper Express Models
Color Processing	On-camera in YUV or RGB format, or on-PC in Raw format
Digital Interface	Two 9-pin IEEE-1394b interfaces for camera control and video data transmission
Transfer Rates	800 Mbit/s
GPIO	8-pin Hirose HR25 GPIO connector for power, trigger, strobe, PWM, and serial I/O, 1 opto-isolated input, 1 opto-isolated output, 2 bi-directional I/O pins
External Trigger Modes	IIDC Trigger Modes 0, 1, 3, 14, and 15
Synchronization	Via external trigger, the IEEE 1394b bus, or free running
Shutter	Global Shutter
	Automatic/Manual/One-Push/Extended Shutter modes
	0.04 ms to >1.5 seconds (extended shutter mode)
Image Buffer	32 MB frame buffer
Memory Channels	2 memory channels for custom camera settings
Flash Memory	1 MB
Dimensions	44 mm x 29 mm x 58 mm excluding lens holder, without optics (metal case)
Mass	86 grams (without optics or tripod mounting bracket)
Power Consumption	8 to 30 V, ~5 W, via GPIO or 1394b interface
Camera Specification	IIDC v1.32
Camera Control	via FlyCapture SDK, CSRs, or third party software
Camera Updates	In-field firmware updates
Lens Mount	C-mount
Operating Temperature	0° to 40°C
Storage Temperature	-30° to 60°C
Emissions Compliance	CE, FCC, RoHS
Operating System	Vista SP1, Windows 7
Warranty	Three years

1.2.1 GX-FW-10K3M (Mono) Imaging Performance

Specification	Mode 0
Full Well Depth	21700 e- at zero gain
Dynamic Range	63 dB
Read Noise	15.3 e- at zero gain
Measurements taken at maximum resolution	

Quantum Efficiency	
Peak QE Wavelength	460 nm
Peak QE Value	51%

Figure 1.1: GX-FW-10K3M Quantum Efficiency

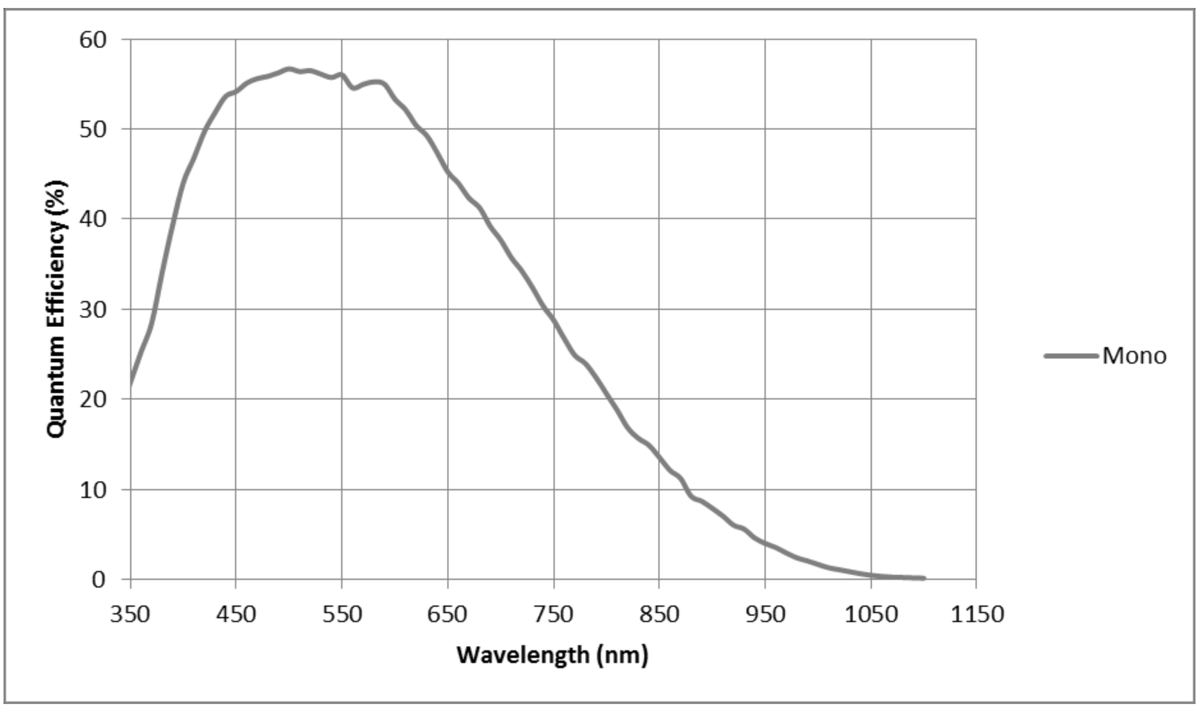


1.2.2 GX-FW-28S5M (Mono) Imaging Performance

Specification	Mode 0
Full Well Depth	16100 e- at zero gain
Dynamic Range	65 dB
Read Noise	9.6 e- at zero gain
Measurements taken at maximum resolution	

Quantum Efficiency	
Peak QE Wavelength	500 nm
Peak QE Value	57%

Figure 1.2: GX-FW-28S5M Quantum Efficiency

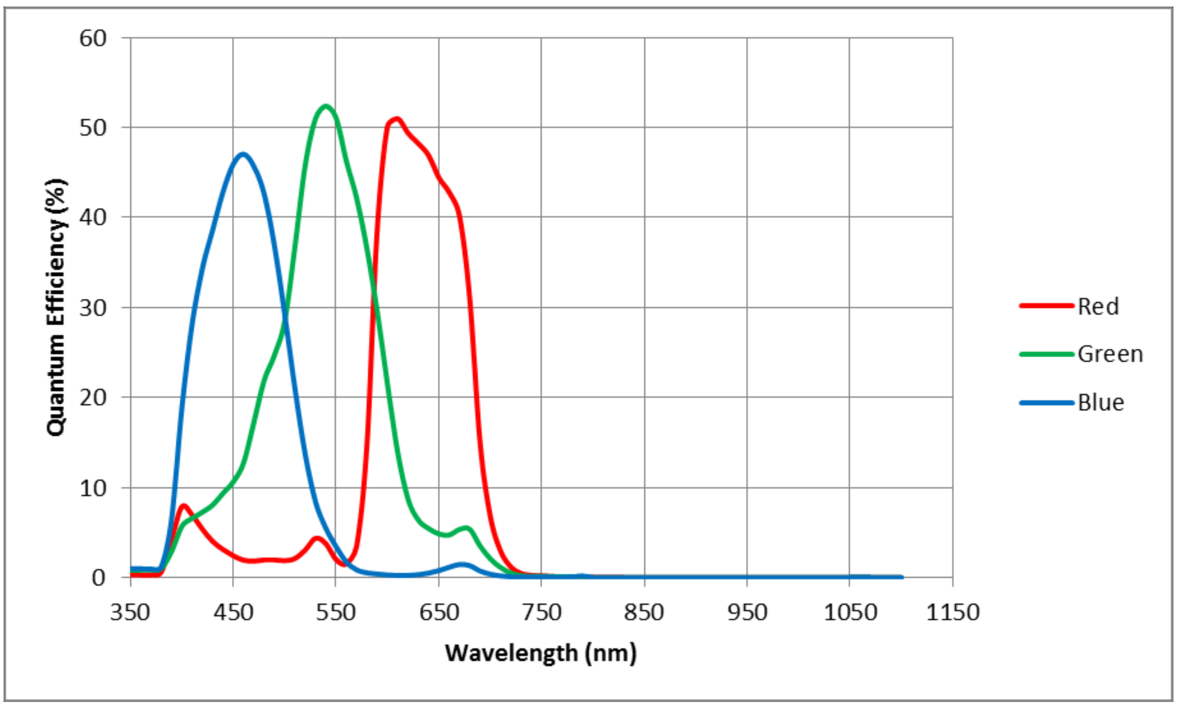


1.2.3 GX-FW-28S5C (Color) Imaging Performance

Specification	Mode 0
Full Well Depth	15300 e- at zero gain
Dynamic Range	65 dB
Read Noise	8.9 e- at zero gain
Measurements taken at maximum resolution	

Quantum Efficiency	
Peak QE Wavelength	Red 610 nm, Green 540 nm, Blue 460 nm
Peak QE Value	Red 51%, Green 52%, Blue 47%

Figure 1.3: GX-FW-28S5C Quantum Efficiency

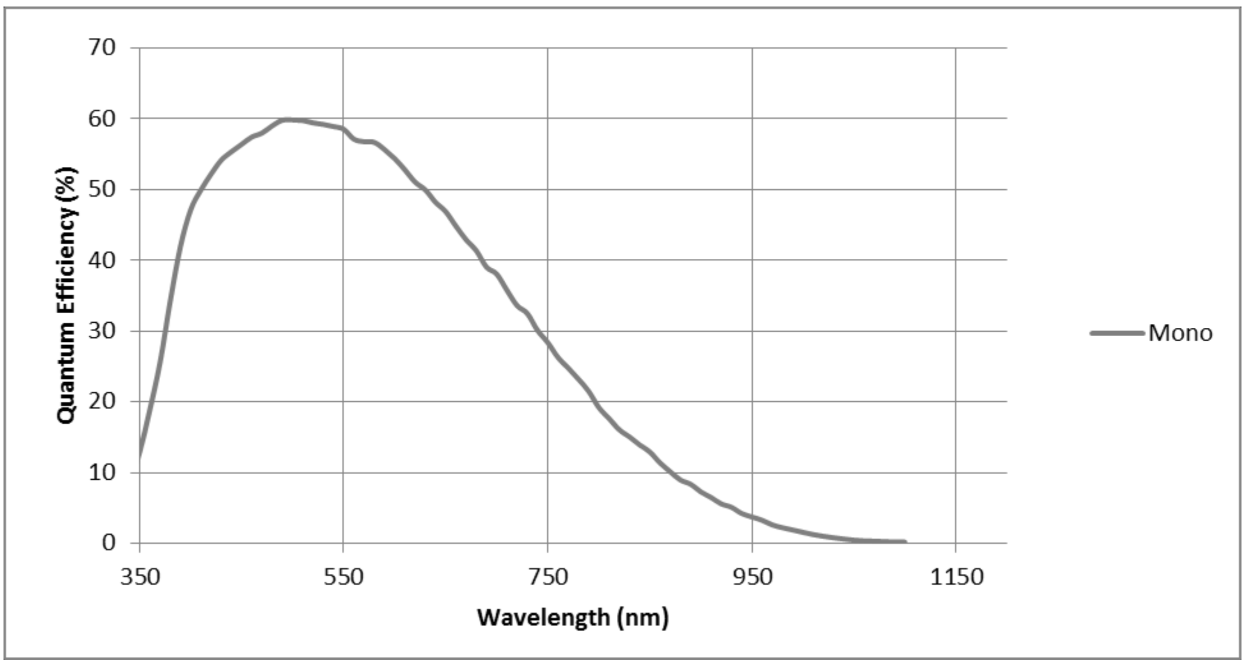


1.2.4 GX-FW-60S6M (Mono) Imaging Performance

Specification	Mode 0
Full Well Depth	16000 e- at zero gain
Dynamic Range	64 dB
Read Noise	10.0 e- at zero gain
Measurements taken at maximum resolution	

Quantum Efficiency	
Peak QE Wavelength	500 nm
Peak QE Value	59%

Figure 1.4: GX-FW-60S6M Quantum Efficiency

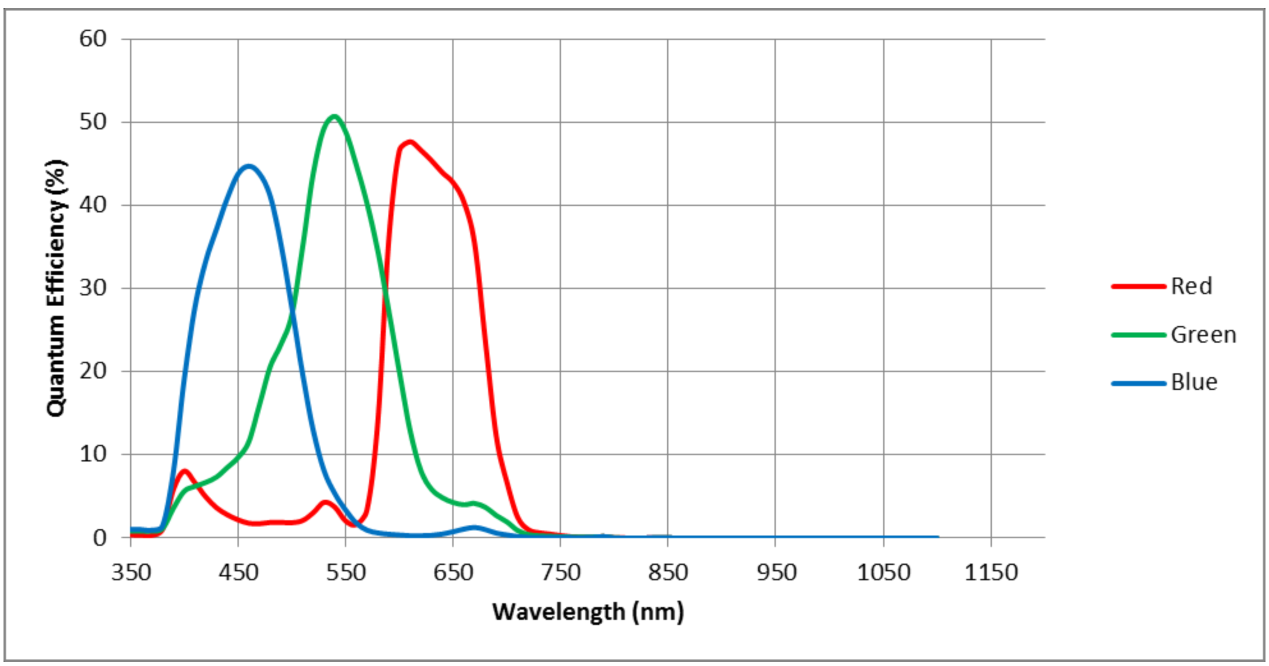


1.2.5 GX-FW-60S6C (Color) Imaging Performance

Specification	Mode 0
Full Well Depth	15500 e- at zero gain
Dynamic Range	64 dB
Read Noise	9.8 e- at zero gain
Measurements taken at maximum resolution	

Quantum Efficiency	
Peak QE Wavelength	Red 610 nm, Green 530 nm, Blue 460 nm
Peak QE Value	Red 47%, Green 49%, Blue 45%

Figure 1.5: GX-FW-60S6C Quantum Efficiency



1.2.6 Grasshopper Express Mono Camera Comparison

Figure 1.6: GX-FW Mono Models Quantum Efficiency

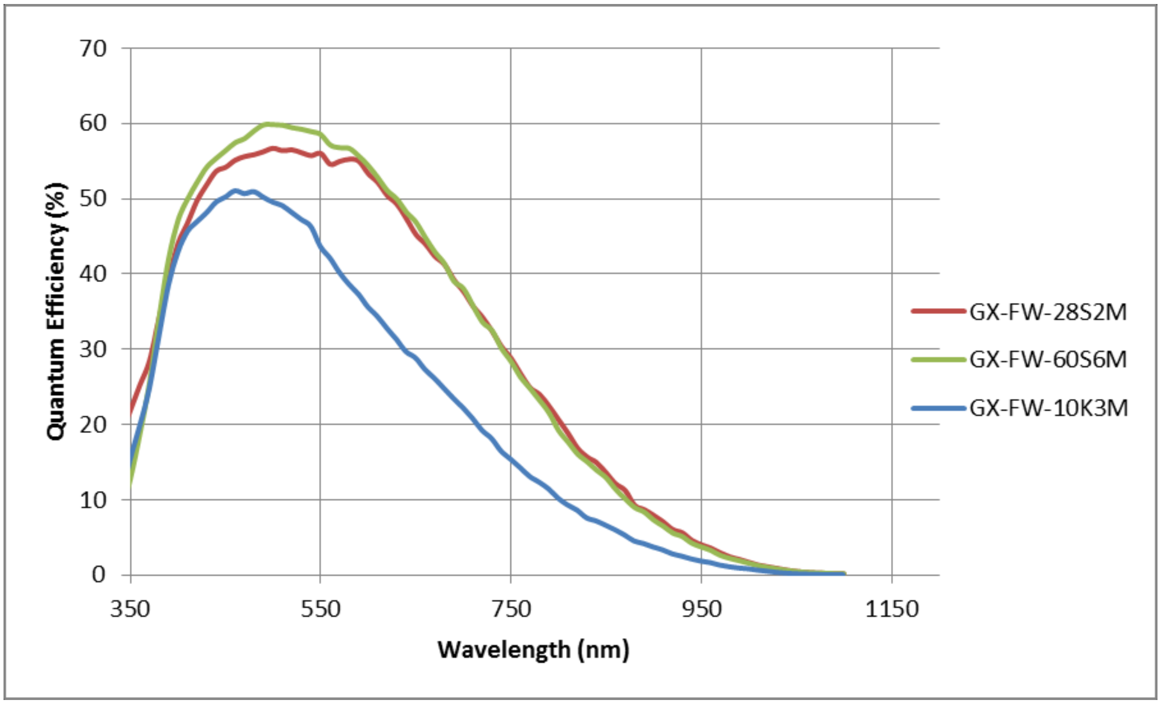
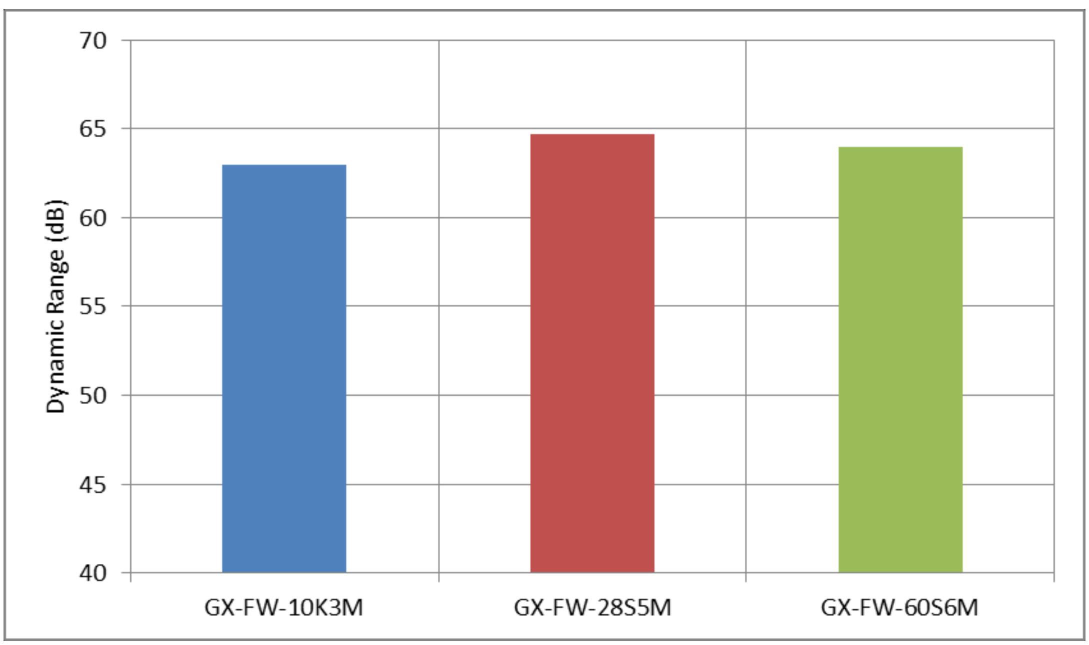
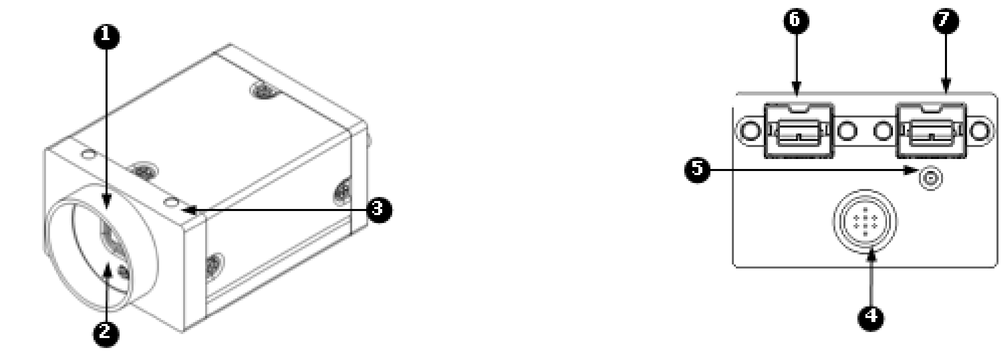


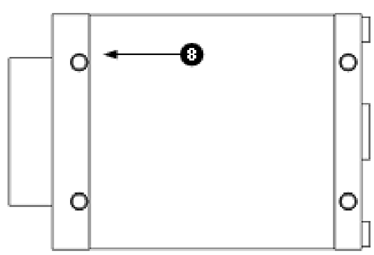
Figure 1.7: GX-FW Mono Models Dynamic Range



1.3 Grasshopper Express Mechanical Properties

1.3.1 Physical Description





1. **Lens holder (C-mount)** (page 11)
2. **Glass/IR filter system** (page 13)
3. **M3 x 0.5 mounting holes** (page 12)
4. **GPIO** (page 37)
5. **Status LED** (page 121)
6. **IEEE-1394 connector** (page 15)
7. **IEEE-1394 connector**
8. **M3 x 0.5 mounting holes** (page 12)

Figure 1.8: Camera Physical Description

1.3.2 Camera Dimensions

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

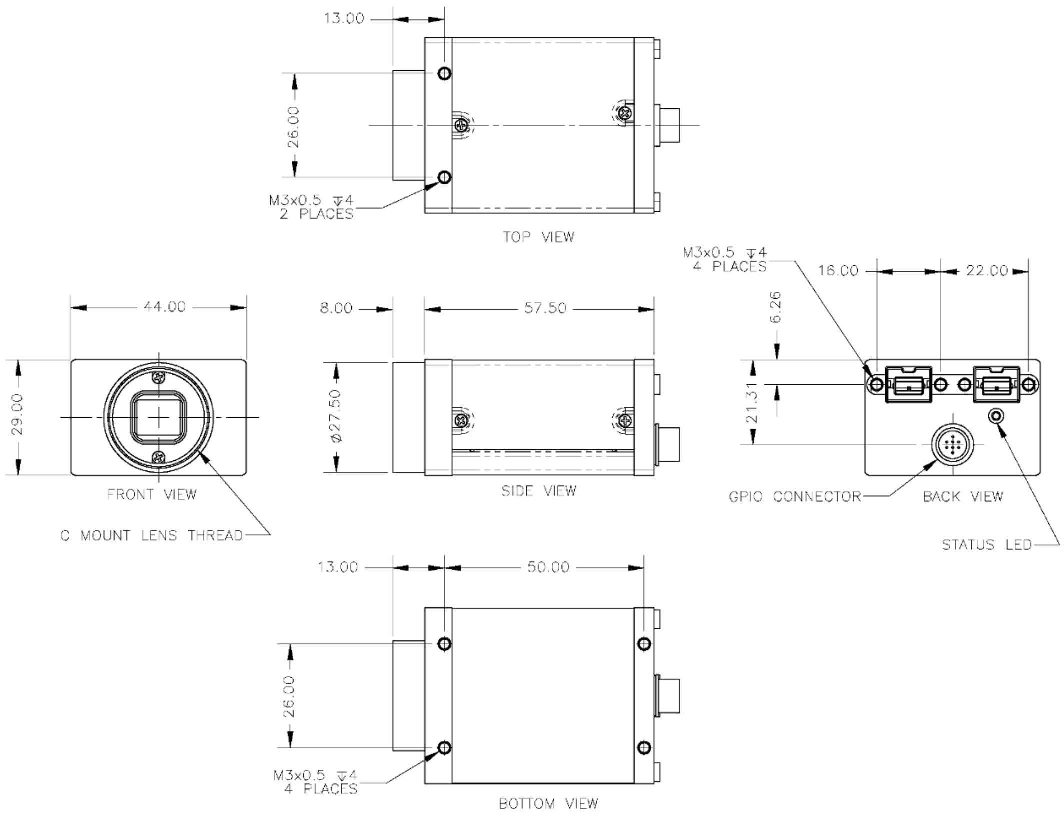


Figure 1.9: Camera Dimensional Diagram

1.3.3 Tripod Adapter Dimensions

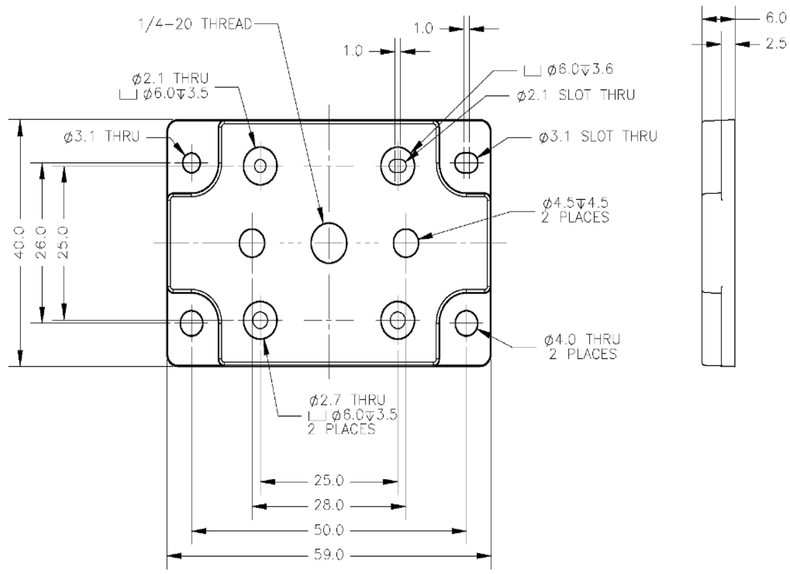


Figure 1.10: Tripod Adapter Dimensional Diagram

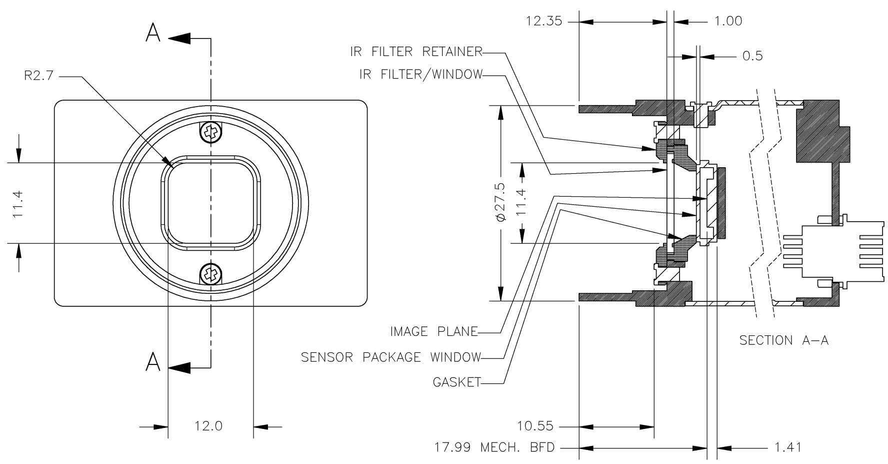
1.3.4 Lens Mounting

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 C-mount lenses. Correct focus cannot be achieved using a CS-mount lens on a C-mount camera.



1.3.4.1 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 sensor package for the GX-FW-10K3M is 0.76 mm.

The resulting BFD is 17.99 mm.

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

1.3.5 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

1.3.6 Mounting with the Case or Mounting Bracket

Using the Case

The case is equipped with the following mounting holes:

- Two (2) M3 x 0.5 mm mounting holes on the top of the case
- Four (4) M3 x 0.5mm mounting holes on the bottom of the case that can be used to attach the camera directly to a custom mount or to the tripod mounting bracket

Using the Mounting Bracket

The tripod mounting bracket is equipped with four (4) M3 mounting holes. For more information, see Tripod Adapter Dimensions on page 11.

1.3.7 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.

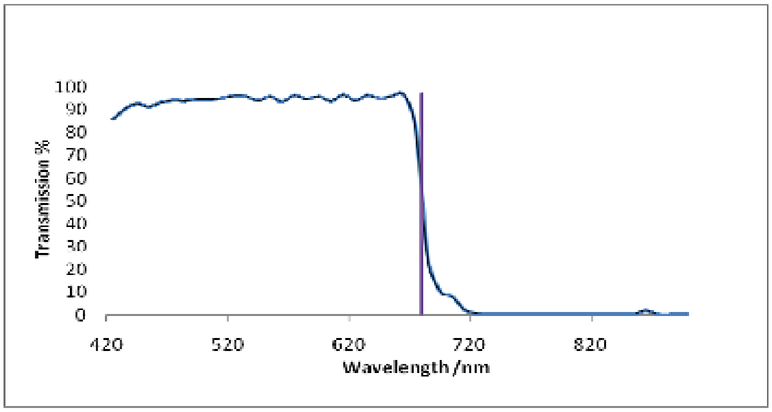


Figure 1.11: IR filter transmittance graph

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
Dimensional Tolerance	+/-0.1 mm
Coating Filters	Scott D 263 T

For more information, see Dust Protection on previous page.

Related Knowledge Base Articles

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

1.4 Analog-to-Digital Conversion

The camera sensor incorporates an A/D converter to digitize the images produced by the CCD. The 14-bit conversion produces 16,384 possible digital image values between 0 and 65,520. Across a 2-byte data format, the two unused bits are padded with zeros. The two least significant bits are always zero.

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

Resolution	14-bit, 65 MHz
Pixel Gain Amplifier	-3 dB to 6 dB, 3 dB steps
Variable Gain Amplifier	6 dB to 42 dB, 10-bit
Black Level Clamp	0 LSB to 1023 LSB, 1 LSB steps

1.5 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. It is recommended to run a grounding wire directly from the case or mounting bracket of the device to the PC motherboard case. This provides a path for the static discharge and should result in reliable operation.
- 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

1.5.1 Case Temperature and Heat Dissipation

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

The camera is equipped with an on-board temperature sensor. It allows you to obtain the temperature of the camera board-level components. The sensor measures the ambient temperature within the case. This feature can be accessed using the TEMPERATURE register 82Ch (page 27).

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 high data rate video 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.

1.6 Camera Interface and Connectors

1.6.1 IEEE-1394b Connector

The camera has two standard 9-pin IEEE-1394b bilingual connectors (pin configuration shown below) for data transmission, camera control and powering the camera. Only one connector can be active at a time. For more detailed information, consult the IEEE-1394b Standard document available from www.1394ta.org.

For more information about powering the camera, see Powering the Camera on page 24

Related Knowledge Base Articles

Title	Article
Key features and benefits of the IEEE-1394b standard	Knowledge Base Article 206

While this camera is an IEEE-1394b device, it is backward compatible with the IEEE-1394a 400Mb/s standard, and can therefore be connected to any 1394a OHCI host adapter using a 9- to 6-pin cable.

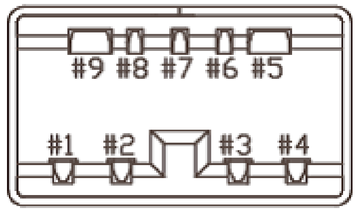


Figure 1.12: IEEE-1394b connector pin configuration (as shown looking at the rear of the camera)

Table 1.2: IEEE-1394b connector pin configuration

Pin	Signal Name	Comment
1	TPB-	Twisted Pair B (Minus)
2	TPB+	Twisted Pair B (Plus)
3	TPA-	Twisted Pair A (Minus)
4	TPA+	Twisted Pair A (Plus)
5	TPA (R)	Twisted Pair A (Reference Ground)
6	V_G	Power (Ground)
7	SC	Status Contact (Reserved for Future Use)
8	V_P	Power (Voltage)
9	TPB (R)	Twisted Pair B (Reference Ground)

1.6.1.1 Daisy Chaining

As the camera has 2 connectors, it is possible to connect multiple cameras and/or hubs to a single host controller. However, the maximum bandwidth available for all cameras is still restricted to 800 Mbps on a 1394b bus or 400 Mbps on a 1394a bus.

1.6.2 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).

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

Related Knowledge Base Articles

Title	Article
Differences to consider when selecting an IEEE-1394 PCI/PCMCIA host adapter card	Knowledge Base Article 146

1.6.3 Interface Cables

Standard, shielded twisted pair copper cables must be used for connections between 1394 nodes, such as from the camera to a PCI card, or a PCI card to a hub.

Related Knowledge Base Articles

Title	Article
Extending the distance between a PGR camera and the controlling host system	Knowledge Base Article 197

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

1.6.4 General Purpose Input/Output (GPIO)

The camera has an 8-pin GPIO connector on the back of the case; refer to the diagram below for wire color-coding. The connector is a Hirose HR25 8 pin connector (Mfg P/N: HR25-7TR-8SA). Male connectors (Mfg P/N: HR25-7TP-8P) can be purchased from Digikey (P/N: HR702-ND).

Diagram	Pin	Function	Description
	1	I0	Opto-isolated input (default Trigger in)
	2	O1	Opto-isolated output
	3	IO2	Input/Output/serial transmit (TX)
	4	IO3	Input/Output/serial receive (RX)
	5	GND	Ground for bi-directional IO, V_{EXT}, +3.3 V pins
	6	OPTO_GND	Ground for opto-isolated IO pins
	7	V_{EXT}	Allows the camera to be powered externally
	8	+3.3 V	Power external circuitry up to 150 mA

Point Grey sells a 12 V wall-mount power supply equipped with a HR25 8-pin GPIO wiring harness for connecting to the camera (**Part No. ACC-01-9006**). For more information, see the miscellaneous product accessories page on the Point Grey website.

For more information on camera power, see Powering the Camera on page 24

For more information on configuring input/output with GPIO, see Input/Output Control on page 37.

For details on GPIO circuits, see GPIO Electrical Characteristics on page 57.

2 Getting Started with Grasshopper Express

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
Vista SP1, Windows 7	2.0 GHz or equivalent	2 GB	AGP128 MB	IEEE 1394b	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:

- FireWire cable (on page 16)
- 8-pin GPIO connector (on page 37)
- C-mount Lens (on page 11)
- Interface card (on page 16)

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

Related Knowledge Base Articles

Title	Article
Does Point Grey sell FireWire/IEEE-1394 and digital camera accessories?	Knowledge Base Article 131

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

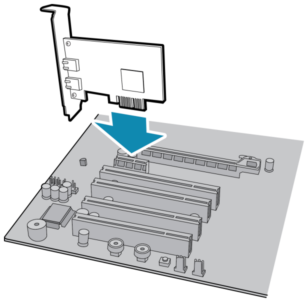
To access the downloads resources you must have a downloads account.

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.

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

In most cases, the Windows IEEE-1394 drivers will be automatically installed for the card, with no user input required. However, in some cases the **Found New Hardware Wizard** will appear. Follow the prompts given by the Wizard to install the card.

Open the Windows Device Manager. Ensure the card is properly installed under **IEEE 1394 Bus host 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. Under **FlyCapture v2x**, 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.

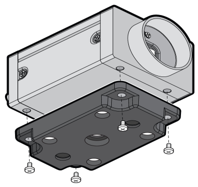
In the **Interface Driver Selection** dialog, select the **I will use FireWire cameras**.

This selection ensures the Point Grey pgrcam and FirePRO drivers are installed and enabled.

To uninstall or reconfigure the driver at any time after setup is complete, use the DriverControlGUI (page 20).

2.3 Installing Your Camera

1. Install the Tripod Mounting Bracket



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

2. Attach a Lens

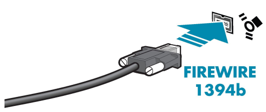
Unscrew the dust cap from the C-mount lens holder to install a lens.

3. Plug in the GPIO connector

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

The wiring harness must be compatible with a Hirose HR25 8-pin female GPIO connector.

4. 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.

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 FlyCap2 program: **Start-> Point Grey Research->FlyCapture2-> FlyCap2**
   The FlyCap2 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 FlyCapture2 SDK (see Configuring Camera Setup below).

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 on page 34.

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-->Point Grey Research-->FlyCapture2-->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.

2.5 Controlling the Camera

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

- FlyCapture2 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.

2.5.1 Using FlyCapture

The user can monitor or control features of the camera through FlyCapture API examples provided in the FlyCapture SDK, or through the FlyCap2 Program.

FlyCap2 Program

The FlyCap2 application is a generic, easy-to-use streaming image viewer included with the FlyCapture2 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.

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.

2.5.2 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.

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

2.5.2.1 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 2.1: 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.

2.5.2.2 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.

2.5.2.3 Using the Inquiry Registers

The camera provides a series of inquiry registers, which allow you to reference basic information about camera features. For information about the following inquiry registers, see:

- *Inquiry Registers for Basic Functions and Feature Presence* (page 138): To determine if a particular function or feature is available on the camera.
- *Inquiry Registers for Feature Elements* (page 141): To determine if elements of a particular feature are available on the camera.
- *Video Format, Mode and Frame Rate Inquiry Registers* (page 129): To determine which standard video format, modes and frame rates are available on the camera.

The following additional inquiry registers are also available:

- *Inquiry Registers for Custom Video Modes* (page 144)
- *Inquiry Registers for Strobe Output* (page 45)
- *Inquiry Registers for Serial I/O* (page 53)
- *Inquiry Registers for Lookup Table Functionality* (page 103)

2.5.2.4 Using the 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.

For more information, see Absolute Value Registers on page 147.

2.5.3 Third-Party Software Applications

The following knowledge base article provides a chart detailing Point Grey camera compatibility with third-party software development kits, applications, camera drivers, and integrated development environments (IDEs).

The chart is meant to act as a general guide toward Point Grey camera compatibility with third- party SDKs, applications, camera drivers, and IDEs. While many Point Grey customers have successfully used many of the following products, we do not officially recommend any product or manufacturer over another. Point Grey does not support any of the following pieces of software. Therefore, any questions regarding this software should be directed to the manufacturer. In addition, Point Grey has not performed extensive testing on any of the following pieces of software, and therefore cannot comment on any errors that occur as a result of using such software.

Related Knowledge Base Articles

Title	Article
PGR Imaging Products compatibility with third-party software and drivers	Knowledge Base Article 152

3 General Camera Operation

3.1 Powering the Camera

The 9-pin 1394 connector connects to a standard IEEE-1394 (FireWire) 9-pin cable and provides a power connection between the camera and the host computer.

The power consumption specification is: 8 to 30 V, ~5 W, via GPIO or 1394b interface.

Some systems - such as laptop computers or those with several FireWire devices connected - require an external power supply to power the camera.

Related Knowledge Base Articles

Title	Article
My laptop's IEEE-1394 port or PCMCIA card doesn't supply power to my camera	Knowledge Base Article 93

Power can also be provided through the GPIO interface. For more information, see General Purpose Input/Output (GPIO) on page 37. The camera selects whichever power source is supplying a higher voltage.

Point Grey sells a 12 V wall-mount power supply equipped with a HR25 8-pin GPIO wiring harness for connecting to the camera (**Part No. ACC-01-9006**). For more information, see the miscellaneous product accessories page on the Point Grey website.

If isochronous transmission (page 74) 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 will revert to its default factory settings, or if applicable, the last saved memory channel. For more information, see User Memory Channels on page 28.

3.1.1 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 (outside IIDC specification)), 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

3.2 Device Information

Information about the camera 's hardware, status and monitoring is available.

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.

Voltage—This allows the user to access and monitor the input as well as several of the internal voltages of the cameras.

Current—This allows the user to access and monitor the current consumption of the camera.

Temperature—Allows the user to get the temperature of the camera board-level components. For cameras housed in a case, it is the ambient temperature within the case. For more information about camera temperature, see Case Temperature and Heat Dissipation on page 14.

Camera Power—Allows the user to power up or power down 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.

3.2.1 SERIAL_NUMBER: 1F20h

Format:

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

3.2.2 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

3.2.3 SENSOR_BOARD_INFO: 1F28h

The interpretation of this register varies depending on the camera type, as defined in the MAIN_ BOARD_INFO register 0x1F24 (page 26). Read MAIN_ BOARD_INFO to determine how to use the Sensor_Type_x fields.

Format:

Field	Bit	Description
Sensor_Type_1	[0-11]	tbd
Minor_Board_Rev	[12-15]	Internal use
Reserved	[16-27]	Reserved
Sensor_Type_2	[28-31]	tbd

3.2.4 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

Offset	Name	Field	Bit	Description
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.

3.2.5 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.

3.2.6 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

3.2.7 PIXEL_CLOCK_FREQ: 1AF0h

Format:

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

3.2.8 HORIZONTAL_LINE_FREQ: 1AF4h

Format:

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

3.3 User Memory Channels

The camera can save and restore settings and imaging parameters via on-board 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.

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

The following camera settings are saved in memory channels.

Frame Rate (including Absolute Value) (page 72)	Image Data Format
Current Frame Rate (page 72)	Image Position and Image Size (page 144)
Current Video Mode (page 72)	Current Video Format (page 72)
Camera Power (page 24)	Frame Information (page 116)
Brightness (including Absolute Value) (page 90)	Trigger Mode (page 79)
Auto Exposure (including Absolute Value and Range) (page 96)	Trigger Delay (including Absolute Value) (page 85)
Sharpness (page 109)	Shutter (including Absolute Value, Auto Shutter Range, and Shutter Delay) (page 91)
White Balance (page 111)	Gain (including Absolute Value and Auto Gain Range) (page 94)
Hue (page 107)	GPIO Pin Modes (page 48)
Saturation (page 106)	GPIO Strobe Modes (page 38)
Gamma (including Absolute Value) (page 100)	GPIO PWM Modes (page 48)
Color Coding ID (page 145)	Format 7 Bytes per Packet

3.3.1 MEMORY_SAVE: 618h

Format:

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

3.3.2 MEM_SAVE_CH: 620h

Format:

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

3.3.3 CUR_MEM_CH: 624h

Format:

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

3.3.4 Memory Channel Registers

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

Register Name	Offset
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
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
FORMAT_7_IMAGE_POSITION	008h
FORMAT_7_IMAGE_SIZE	00Ch
FORMAT_7_COLOR_CODING_ID	010h
FORMAT_7_BYTE_PER_PACKET	044h
UDP_PORT	1F1Ch
DESTINATION_IP	1F34h

3.4 On-Camera Frame Buffer

The camera has 32 MB of memory that can be used for temporary image storage. This may be useful in cases such as:

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

All images pass through the frame buffer mechanism. This introduces relatively little delay in the system because the camera does not wait for a full image to arrive in the buffer before starting transmission but rather lags only a few lines behind.

The user can cause images to accumulate by enabling the frame buffer. This effectively disables the transmission of images in favor of accumulating them in the frame buffer. The user is then required to use the remaining elements of the interface to cause the transmission of the images.

The buffer system is circular in nature, storing only the last 32 MB worth of image data. The number of images that this accommodates depends on the currently configured image size.

The standard user interaction involves the following steps:

1. **Configure the imaging mode.**
   This first step involves configuring the format, mode and frame rate for acquiring images. This can be done by either directly manipulating the registers or using the higher level functionality associated with the software library being used. Depending on the software package, this may involve going so far as to configure the camera, perform bandwidth negotiation and grab an image. In cases where bandwidth is restricted, the user will want to disable transmission and free the bandwidth after the camera is configured.
2. **Enable frame buffer accumulation**
   The second step involves enabling the frame buffer. Enabling this results in images being accumulated in the frame buffer rather than immediately being transmitted.
3. **Negotiate bandwidth with the camera**
   Having accumulated some number of images on the camera, bandwidth will have to be renegotiated if it has not been done already. In most cases, this will involve effectively starting the camera in the imaging mode configured in step (1).
4. **Disable isochronous transmission and enable buffered image transfer**
   To transfer buffered images, isochronous data transmission must be disabled, and transfer data enabled.
5. **Transmit images off of the camera**
   The final step involves setting One Shot/Multi-shot in order to cause the camera to transmit one or more images from the frame buffer over the data interface.

Although it is possible to repeatedly transmit the same image, there is no way to access images that are older than the last image transmitted.

Whether by enabling trigger or disabling isochronous data, switching out of a free running mode leaves the last image transmitted in an undefined state.

The frame buffer is volatile memory that is erased after power cycling. To store images on the camera after power cycling, use Non-Volatile Flash Memory on page 33. Accessing flash memory is significantly slower than accessing the frame buffer, and storage is limited.

3.4.1 IMAGE_RETRANSMIT: 634h

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

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

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

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

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

Format:

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

3.4.2 Example: Retransmitting in Image External Mode Using Registers

There are occasions where it might be beneficial to retransmit an image when in an external trigger mode. Having configured the camera to be running in an external trigger mode, the user can cause the camera to retransmit an image by doing the following:

1. **Read the current state of the IMAGE_RETRANSMIT register 634h:**

Read	634h	00 07 00 00

Reading register 634h indicates that the frame buffer mechanism is currently disabled, and in the current imaging mode, the system is capable of storing up to 7 images.

2. **Enable image hold:**

Write	634h	80 07 00 00

Setting bit 0 of register 634h to 1 enables images to accumulate in the frame buffer.

3. **Enable buffered image transfer:**

Write	634h	C0 07 00 00

Setting bit 1 of register 634h to 1 enables transfer of buffered image data.

4. **Retransmit the last image:**

Write	61Ch	80 00 00 00

Setting bit 0 of register 61Ch to 1 causes the last image to be retransmitted.

5. **Disable buffered image transfer:**

Write	634h	00 07 00 00

Writing 0 to bits 0 and 1 of register 634h disables buffered image hold and transfer, and returns the camera to normal operation.

3.4.3 Example: Storing Images for Later Transmission Using Registers

Again, assuming the camera is configured to run in an external trigger mode:

1. **Read the current state of register 634h:**

Read	634h	00 07 00 00

Again, this value indicates that the frame buffer mechanism is currently disabled, and in the current imaging mode the system is capable of storing up to 7 images.

2. **Enable hold image mode and buffer data transfer:**

Write	634h	C0 07 00 00

Setting bits 0 and 1 of register 634h enables image buffer hold and transfer, resulting in images being accumulated in the frame buffer for later transmission.

3. **Acquire 4 images:**

Write	62Ch	80 00 00 00
Write	62Ch	80 00 00 00
Write	62Ch	80 00 00 00
Write	62Ch	80 00 00 00
Read	634h	C0 07 00 04

Writing to software trigger register 62Ch 4 times causes 4 images to accumulate in the frame buffer. The last 12 bits of register 634h will now indicate that there are 4 images in the frame buffer.

4. **Transmit two images:**

Write	61Ch	80 00 00 00
Write	61Ch	80 00 00 00
Read	634h	C0 07 00 02

Writing 1 to bit 0 of register 61Ch results in a single image being transmitted and the number of images available being decremented by one. After transmitting two images, a subsequent read of the register indicates that there are two images left.

3.5 Non-Volatile Flash Memory

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

Related Knowledge Base Articles

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

3.5.1 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

3.5.2 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

3.6 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

3.6.1 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

3.6.2 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.

3.6.3 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

3.6.4 FIRMWARE_BUILD_DATE: 1F64h

Format:

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

3.6.5 FIRMWARE_DESCRIPTION: 1F68-1F7Ch

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

4 Input/Output Control

4.1 General Purpose Input/Output (GPIO)

The camera has an 8-pin GPIO connector on the back of the case; refer to the diagram below for wire color-coding. The connector is a Hirose HR25 8 pin connector (Mfg P/N: HR25-7TR-8SA). Male connectors (Mfg P/N: HR25-7TP-8P) can be purchased from Digikey (P/N: HR702-ND).

Table 4.1: GPIO pin assignments (as shown looking at rear of camera)

Diagram	Pin	Function	Description
	1	IO	Opto-isolated input (default Trigger in)
	2	O1	Opto-isolated output
	3	IO2	Input/Output/serial transmit (TX)
	4	IO3	Input/Output/serial receive (RX)
	5	GND	Ground for bi-directional IO, V_{EXT}, +3.3 V pins
	6	OPTO_GND	Ground for opto-isolated IO pins
	7	V_{EXT}	Allows the camera to be powered externally
	8	+3.3 V	Power external circuitry up to 150 mA

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

Point Grey sells a 12 V wall-mount power supply equipped with a HR25 8-pin GPIO wiring harness for connecting to the camera (**Part No. ACC-01-9006**). For more information, see the miscellaneous product accessories page on the Point Grey website.

For details on GPIO circuits, see GPIO Electrical Characteristics on page 57.

4.2 GPIO Modes

4.2.1 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 on page 49.

4.2.2 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.*

4.2.3 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 on page 77.

4.2.4 GPIO Mode 3: Strobe

A GPIO pin in GPIO Mode 3 will output 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 below.

4.2.5 GPIO Mode 4: Pulse Width Modulation (PWM)

When a GPIO pin is set to GPIO Mode 4, the pin will output a specified number of pulses with programmable high and low duration. See Pulse Width Modulation (PWM) on page 48.

4.3 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

4.3.1 Example: Setting a GPIO Pin to Strobe (Using the FlyCapture API)

The following FlyCapture 2.x 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.

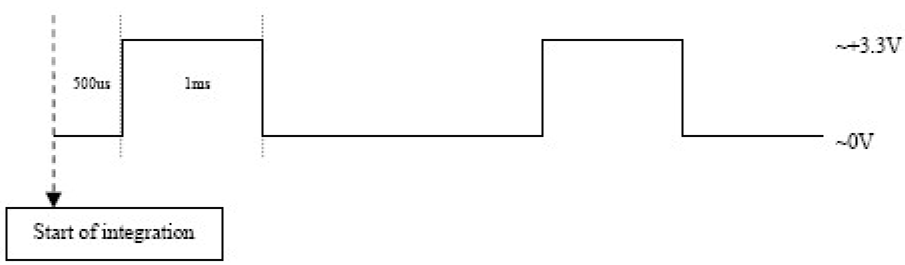
Assuming a `Camera` object `cam`:

```
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);
```

4.3.2 Example: Setting a GPIO Pin to Strobe (Using the Camera Registers)

Consider the following example strobe scenario:

- Desired strobe output pin: GPIO2
- Strobe output characteristics: 500us delay from start of shutter, 1ms high duration (see below)



Determine the Default Output Pins

Electrically, general purpose input/output pins are in one of two states: input or output. In order for a GPIO pin to act as a strobe output source, it must be configured as an output. To determine which of the GPIO pins are outputs by default, get the value of the PIO_DIRECTION register 0x11F8 (page 87). The IOx_Mode fields (bits 0-3) report the current state of the corresponding pin. For example:

`0x11F8 = 0x4000 0000`

4	0	0	0	0	0	0	0	Hex
0100	0000	0000	0000	0000	0000	0000	0000	Binary
0-7		8-15		16-23		24-31		Bits

Each of the first four bits represents the current state of its associated GPIO pin: '0' indicates it is an input/trigger, and '1' indicates it is an output/strobe. In the example above, 0x4 = 0100 in binary, so GPIO1 is configured as an output and GPIO0, GIPIO2 and GPIO3 are inputs.

Set the Desired Pin as an Output

Following the example above, assume we want to configure GPIO2 to be an output. To do this, set the appropriate bit of the PIO_DIRECTION register 0x11F8 (in this case bit 2) to '1'. In the example above, we would therefore do the following register write:

`0x11F8 = 0x6000 0000`

Determine Strobe Support

The next step is to determine whether our desired strobe pin, GPIO2, is capable of outputting a strobe signal. To do this, get the value of the appropriate STROBE_x_INQ register (page 45); in this case, the STROBE_0_INQ register 0x1408. Assuming we have correctly configured GPIO2 to be an output, we should get a value of:

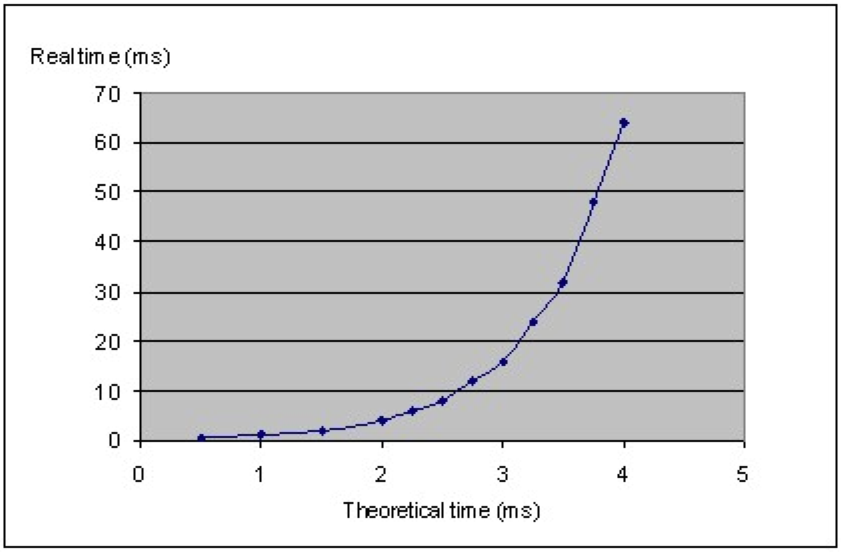
8	E	0	0	0	F	F	F	Hex
1000	1110	0000	0000	0000	1111	1111	1111	Binary
0-7		8-15		16-23		24-31		Bits

Bit 0 is a '1', which confirms that the strobe functionality is present on this GPIO pin. Bit 4 points to the ability to read the value of this feature. Bit 5 indicates the ability to turn the strobe on and off, and bit 6 indicates that we can change the strobe signal polarity. Bits 8-19 are '0', which means the minimum strobe duration is zero. Bits 20-31 are '0xFFF' or 4096 in decimal, so the maximum strobe delay and duration is 4096.

Configure the Desired Pin to Output a Strobe

At this point, GPIO2 is set as an output pin and we know it can be a strobe signal source. Now, we need to enable it as a strobe source by "turning it on" using the GPIO pin's STROBE_x_CNT register (page 45).

Continuing our example, the desired strobe pin is GPIO2. Therefore, we want to look at the STROBE_2_CNT register 0x1508. The values that we enter in the *Delay_Value* and *Duration_Value* fields of this register are determined as follows: for values up to approximately 0x400 (1024 decimal), each value increment is a tick of a 1.024MHz clock. Values between 0x401 and 0xFF become non-linear in the manner shown in the figure below:



Duration_Value/Delay_Value	Real Time (ms)
0x050	0.078
0x200	0.5
0x400	1
0x600	2
0x800	4
0x900	6
0xA00	8
0xB00	12
0xC00	16
0xD00	24
0xE00	32
0xF00	48
0xFFF	63.93

For example, to achieve a 500us delay and 1ms duration we calculate:

Delay_Value = 0.0005s * 1024000Hz = 512 = 0x200

Duration_Value = 0.001s * 1024000Hz = 1024 = 0x400

To finish configuring GPIO2 to output a strobe pulse of 500us delay from the start of integration and 1ms high duration (high active output), we make the following final register write:

```
0x1508 = 0x8320 0400
```

4.3.3 Example: Setting GPIO Pins to Output a Strobe Pattern

Consider the following example strobe pattern scenario for two cameras on the same bus:

Parameter	Value
Frame rate	15Hz
Shutter (integration) time	Manual mode, 15ms
Strobe output pin	GPIO2
Strobe output characteristics	High active output, 1ms duration
Strobe output pattern (Cam_A)	Strobe every three (3) frames. The strobe should occur on the third frame. Effective strobe frequency = 15 / 3 = 5Hz.
Strobe output pattern (Cam_B)	Strobe every two (2) frames. The strobe should occur on the second frame. Effective strobe frequency = 15 / 2 = 7.5Hz.

This configuration is outlined below. Multiple cameras on the same bus are automatically synchronized to each other, so with this configuration the strobes of the two cameras should be synchronized at every sixth frame.

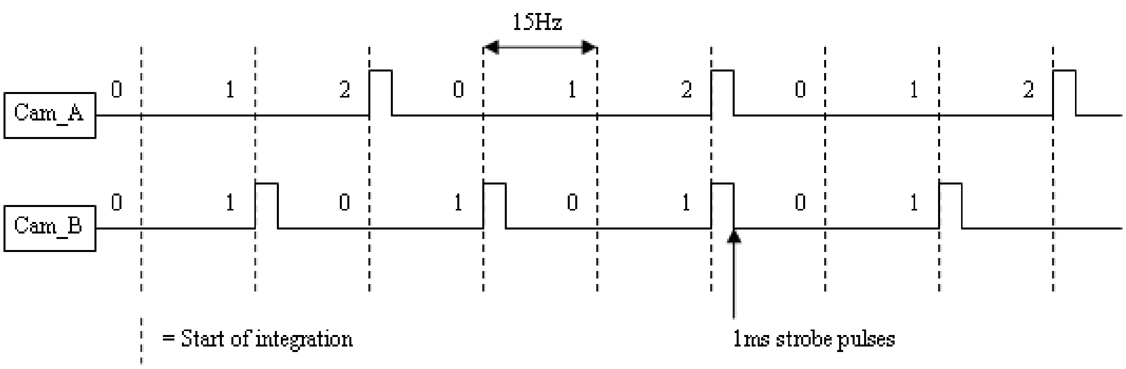


Figure 4.1: Example of multiple camera strobe pattern synchronization

4.3.3.1 Start the Camera

Using a separate instance of the FlyCap demo program for each camera, begin grabbing images at the frame rate specified above. Put the shutter in manual mode and set the shutter time to be the same for each camera.

4.3.3.2 Configure the Desired Pins to Output a Strobe

See Example: Setting a GPIO Pin to Strobe (Using the Camera Registers) on page 39 for instructions on configuring each camera to output a 1ms pulse off a GPIO pin.

4.3.3.3 Configure the Strobe Pattern Period

The programmable strobe pattern period is controlled by the *Count_Period* field of the GPIO_STRPAT_CTRL register 0x110C (page 46). The valid values for *Count_Period* are 1 – 16. By default (normal strobe mode), the *Count_Period* is 1 and the camera strobes on every frame.

At the start of integration for each image, the period counter is incremented and the *Current_Count* read-only field is updated. In the example above, Cam_A has a period of three (3). *Current_Count* therefore counts from 0 up to 2, then wraps around back to 0 to start the next period.

To configure the strobe periods for the two cameras in our example we need to set the *Count_ Period* by writing the following to register 0x110C:

```
(Cam_A) 0x110C = 0x8000 0300
```

8	0	0	0	0	3	0	0	Hex
1000	0000	0000	0000	0000	0011	0000	0000	Binary
0-7		8-15		16-23		24-31		Bits

```
(Cam_B) 0x110C = 0x8000 0200
```

8	0	0	0	0	2	0	0	Hex
1000	0000	0000	0000	0000	0010	0000	0000	Binary
0-7		8-15		16-23		24-31		Bits

Bit [0] indicates the Presence of the strobe pattern feature; awrite to this bit is therefore a NoOp. Bits [19-23] control the *Count_Period*, which is 3 for Cam_A and 2 for Cam_B.

Reading this register, depending on how quickly the read returns, should show the *Current_ Count* value cycling through its range of values: 0, 1, 2 for Cam_A; and 0, 1 for Cam_B.

4.3.3.4 Define the Strobe Pattern

Following the example above, we want Cam_A to strobe on the third frame only (*Current_Count* = 2) and Cam_B to strobe on the second frame only (*Current_Count* = 1). To do this, we need to configure the *Enable_Mask* in the GPIO_ STRPAT_MASK_PIN_2 register 0x1138 (page 46).

In general, when the *Current_Count* equals N the GPIO pin will only output a strobe if Bit [N] of the *Enable_Mask* is set to '1'. By default, all of the bits in the *Enable_Mask* are set to '1' (a read of 0x1138 would display 0x8000FFFF) so that the strobe occurs for every frame.

In our example above, we want Cam_A to strobe when the *Current_ Count* equals 2. We therefore need to set Bit [2] of the *Enable_ Mask*, which is actually Bit [18] of the GPIO_STRPAT_MASK_PIN_2 register, to '1'. To ensure a strobe does not occur when *Current_Count* is equal to 0 or 1, we need to set Bits [0-1] to '0'. We want Cam_B to strobe when the count equals 1, so we need to set Bit [1] (Bit [17] of register 0x1138) to '1' and Bit [0] to '0'. Our register writes would therefore look like this:

```
(Cam_A) 0x1118 = 0x8000 3FFF
```

8	0	0	0	3	F	F	F	Hex
1000	0000	0000	0000	0011	1111	1111	1111	Binary
0-7		8-15		16-23		24-31		Bits

```
(Cam_B) 0x1118 = 0x8000 7FFF
```

8	0	0	0	7	F	F	F	Hex
1000	0000	0000	0000	0111	1111	1111	1111	Binary
0-7		8-15		16-23		24-31		Bits

At this point, the strobe pulses coming off each camera should look very similar to those in Example of multiple camera strobe pattern synchronization on page 42. If they are not aligned correctly, simply adjust the *Enable_Mask* until the pulses are synchronized.

4.3.3.5 Matching Strobe Pulses to Images

It may be useful to know whether a given image would have a strobe pulse associated with it. For example, the strobe may be used to turn on a small LED or lighting system that illuminates the scene for a specific image. It may be useful to know whether a grabbed image, which can be accessed in memory, is one that should be illuminated. In the same way, it is also useful to know at what point in the pattern an image is grabbed, which might allow the user to make various camera changes in anticipation of the next strobe.

The easiest way to accomplish this is to embed the value of the 32-bit GPIO_STRPAT_CTRL register 0x110C (page 46) into the image by setting Bit [24] of the FRAME_INFO register 0x12F8 (page 118). This register allows the user to control the types of frame-specific information that are embedded into the first several pixels of the image.

Following the example above, we would therefore write the following for both cameras:

```
0x12F8 = 0x8000 0080
```

8	0	0	0	0	0	8	0	Hex
1000	0000	0000	0000	0000	0000	1000	0000	Binary
0-7		8-15		16-23		24-31		Bits

Once this is done, the first four pixels (or 4 bytes of image data) represent the 32-bit (4-byte) value of the GPIO_STRPAT_CTRL register. From this, the application can parse the image data to look for the *Current_Count* Bits [28-31] to determine whether or not the current image should have a strobe associated with it.

Following the example above, the following FlyCapture 2.0 API code snippet applies to Cam_A. The snippet assumes an `Image` object `RawImage`, and that `strobePattern` is the only information being embedded in the image:

```
unsigned char* data;
unsigned char ucCurrentCount;
data = rawImage.GetData();
ucCurrentCount = data[3]; // Byte[4] or bits[28-31] of register 0x110C
if(ucCurrentCount == 2)
{
//a strobe should have occurred for this image - Do something
}
```

4.3.4 Strobe Signal Output Registers

This section describes the control and inquiry registers for the Strobe Signal functionality.

4.3.4.1 Strobe Output 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)*

Format:

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
Base + 0h	STROBE_CTRL_ INQ	Strobe_0_Inq	[0]	Presence of strobe 0 signal
		Strobe_1_Inq	[1]	Presence of strobe 1 signal
		Strobe_2_Inq	[2]	Presence of strobe 2 signal
		Strobe_3_Inq	[3]	Presence of strobe 3 signal
		-	[4-31]	Reserved
Base + 100h	STROBE_0_INQ	Presence_Inq	[0]	Presence of this feature
			[1-3]	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 signal polarity
			[7]	Reserved
		Min_Value	[8-19]	Minimum value for this feature control
		Max_Value	[20-31]	Maximum value for this feature control
Base + 104h	STROBE_1_INQ	Same definition as Strobe_0_Inq		
Base + 108h	STROBE_2_INQ	Same definition as Strobe_0_Inq		
Base + 10Ch	STROBE_3_INQ	Same definition as Strobe_0_Inq		

Offset	Name	Field	Bit	Description
Base + 200h	STROBE_0_CNT	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 this function 0: OFF, 1: ON If this bit = 0, other fields will be read only. *When ON, strobe signals continue to output after the camera stops streaming images. To stop strobe output, this bit must be explicitly turned OFF.*
		Signal_Polarity	[7]	Select signal polarity If Polarity_Inq = 1: Read to get strobe output polarity Write to change strobe output polarity If Polarity_Inq = 0, then Read only 0: Low active output, 1: High active output
		Delay_Value	[8-19]	Delay after start of exposure until the strobe signal asserts
		Duration_Value	[20-31]	Duration of the strobe signal A value of 0 means de-assert at the end of exposure, if required.
Base + 204h	STROBE_1_CNT	Same definition as Strobe_0_ Cnt		
Base + 208h	STROBE_2_CNT	Same definition as Strobe_0_Cnt		
Base + 20Ch	STROBE_3_CNT	Same definition as Strobe_0_Cnt		

4.3.4.2 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

Field	Bit	Description
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.

4.3.4.3 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
2	GPIO_STRPAT_MASK_PIN_2	1138h
3	GPIO_STRPAT_MASK_PIN_3	1148h

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

4.3.4.4 GPIO_XTRA: 1104h

The GPIO_XTRA register controls when a strobe starts: relative to the start of integration (default) or relative to the time of an asynchronous trigger.

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

4.4 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.

4.4.1 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
2	GPIO_CTRL_PIN_2	1130h
3	GPIO_CTRL_PIN_3	1140h

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

Field	Bit	Description
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
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

4.4.2 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
2	GPIO_XTRA_PIN_2	1134h
3	GPIO_XTRA_PIN_3	1144h

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)

4.5 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.

Related Knowledge Base Articles

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

SIO Buffers

Both the transmit and receive buffers are implemented as circular buffers that may exceed the 255 byte maximum.

- The transmit buffer size is 512 B.
- The receive buffer size is 8 KB.

Block reads and writes are both supported. Neither their length nor their address have to be 32-bit aligned or divisible by 4.

4.5.1 Serial Output Transaction (Transmitting Data)

A general overview of the steps for a serial output transaction, where the camera is transmitting data to a receiving serial port, is as follows:

1. In TRANSMIT_BUFFER_STATUS_CONTROL register 7000Ch, read the available data space of the current transmit buffer *TBUF_ST* field.
2. Write characters to the SIO_DATA_REGISTER 70100h.
3. In TRANSMIT_BUFFER_STATUS_CONTROL register 7000Ch, write the valid output data length to the *TBUF_CNT* field to start transmit.
4. To output more characters, repeat step 1.

4.5.1.1 Example: Transmitting Characters to a PC

This example describes how to send four (4) characters from the camera to the serial port on a PC. Microsoft's HyperTerminal program *(Start Menu > All Programs > Accessories > Communications)* is used to display the characters received from the camera. The process detailed by the table below involves the user enabling transmit, verifying that the transmit buffer is ready, writing four characters to the transmit buffer via the data access registers and then verifying that the characters are ready before finally transmitting them.

Step	Action	Register	Input/Expected Output
1. Plug the camera in and start FlyCap.			
2. Open the Camera Control Dialog and select the Register tab.			
3. Get the current baud rate, character length setting, parity setting and stop bit setting.	Get Register	0x70000	0x060800FF ▪ *0x06 = 19200bps* ▪ *0x08 = 8bit, no parity, 1 stop* ▪ *0xFF = 255 byte buffer*

Step	Action	Register	Input/Expected Output
4. Open a HyperTerminal window and create a new connection, setting the COM Port Settings to match the current camera settings obtained in step 3.			
5. Enable the serial output (transmit).	Set Register	0x70004	0x40000000
6. Verify transmit buffer ready.	Get Register	0x70004	0x40800000
7. Send four (4) characters to the output buffer on the camera.	Set Register	0x70100	0x31323334 ▪ *ASCII = 1234*
8. Verify that the transmit buffer is currently storing 4 bytes worth of characters.	Get Register	0x7000C	0xFF040000 ▪ *0xFF = 255 bytes of buffer space remaining* ▪ *0x04 = 4 bytes currently stored and waiting to be transmitted*
9. Send the characters from the output buffer to the PC's serial port.	Set Register	0x7000C	0xFF040000 ▪ *HyperTerminal should echo the characters "1234"*

To send more than four characters, either:

- Repeat steps 7 through 9 above, and send characters in sets of four using register 0x70100; or
- Do a block write of all the characters using registers 0x70104 – 0x701FF (see the FlyCapture API documentation for information on doing block transfers).

Although both types of writes to the transmit buffer may have to be 32-bit aligned, the number of characters transmitted does not. Subsequent writes to the buffer will simply overwrite characters that were not transmitted during a previous transmit.

The actual transmit buffer size may be larger than that reported in step 3 above. See SIO Buffers on previous page. When this is the case, the "buffer space remaining" that is reported in step 8 will not decrease until the actual buffer space remaining is less than 255 bytes.

4.5.2 Serial Input Transaction (Receiving Data)

A general overview of the steps for a serial input transaction, where the camera is receiving data from a transmitting serial port, is as follows:

1. In RECEIVE_BUFFER_STATUS_CONTROL register 70008h, read the valid data size of current receive buffer *RBUF_ST*.
2. Write the input data length to *RBUF_CNT* field.
3. Read received characters from SIO_DATA_REGISTER 70100h.
4. To input more characters, repeat step 1.

4.5.2.1 Example: Receiving Characters from a PC

This example describes how to send four (4) characters from the PC to the camera's serial port. Microsoft's HyperTerminal program (*Start Menu > All Programs > Accessories > Communications*) is used to send the characters received from the camera. The process detailed by the table below involves the user enabling receive, having characters sent to the camera, checking to insure that the receive buffer is ready to be read, verifying that the characters have arrived and then having them transferred to the data access registers before they are read out.

Step	Action	Register	Input/Expected Output
1. Repeat steps 1 to 4 described in *Example: Transmitting Characters to a PC* (page 50)			
2. Enable the serial input (receive).	Set Register	0x70004	0x80000000
3. Verify no receive data framing errors. (page 52)	Get Register	0x70004	0x80000000 ▪ *0x80040000 indicates a receive data framing error, possibly due to a noisy RS-232 line or incorrect baud rate/port settings.* ▪ *0x80020000 indicates a receive data parity error*
4. Send four (4) characters to the input buffer on the camera. For test purposes, type the characters "ABCD" in the HyperTerminal window.			By default, characters will not be displayed in the HyperTerminal window. To echo typed characters to the screen, select File > Properties > Settings tab > ASCII Setup…
5. Verify that the receive data buffer is ready to be read.	Get Register	0x70004	0x80200000
6. Verify that the receive buffer is currently storing 4 bytes worth of characters, which are waiting to be read.	Get Register	0x70008	0x04000000
7. Send four (4) characters from the input buffer to the data access register.	Set Register	0x70008	0x00040000
8. Verify that four (4) characters are ready to be read from the data access register.	Get Register	0x70008	0x00040000
9. Read the four (4) characters from the data access register.	Get Register	0x70100	0x41424344 ▪ *Assumes input was "ABCD"*

To receive more than four characters, either:

- Repeat steps above, and receive characters in sets of four using register 70100h; or
- Do a block read of all of the characters using registers 0x70104 – 0x701FF. For example, if 12 characters were received (0x70008 = 0x0C000000), Set Register 0x70008 to 0x000C0000 and begin reading the bytes starting at 0x70104 (see the FlyCapture API documentation for information on doing block transfers).

Although both types of reads from the receive buffer may have to be 32-bit aligned, the number of characters received does not. Extra characters read will simply be filled with 0's.

The actual receive buffer size may be larger than that reported in step 3 above. See SIO Buffers on page 50.

4.5.3 Transmitting and Receiving Data Simultaneously

Simultaneous transmitting and receiving of data can be achieved in a manner very similar to that illustrated by the previous two examples. The primary difference is that register 70004h must be set to 0xC0000000 to enable both transmit and receive. Once this has been done transmit and receive transactions can be interleaved as may be required by the application.

4.5.4 SIO Control and Inquiry Registers

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

4.5.4.1 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
		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 50 If this value=1, *Buffer_Status_Control* and *SIO_Data_Register* characters 1-3 should be ignored.

Offset	Name	Field	Bit	Description
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
		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

Offset	Name	Field	Bit	Description
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
		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.

Offset	Name	Field	Bit	Description
		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.

4.6 GPIO Electrical Characteristics

Opto-isolated **input** pins require an external pull up resistor to allow triggering of the camera by shorting the pin to the corresponding opto ground (OPTO_GND). Non opto-isolated input pins are internally pulled high using weak pull-up resistors to allow triggering by shorting the pin to GND. Inputs can also be directly driven from a 3.3 V or 5 V logic output.

The inputs are protected from over voltage.

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 7) allows the camera to be powered externally. The voltage limit is 8-30 V, and current is limited to 1 A.

The **+3.3V** pin is fused at 150 mA. External devices connected to Pin 8 should not attempt to pull anything greater than that.

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

4.6.1 GPIO0 (Opto-Isolated Input) Circuit

The figure below shows the schematic for the opto-isolated input circuit.

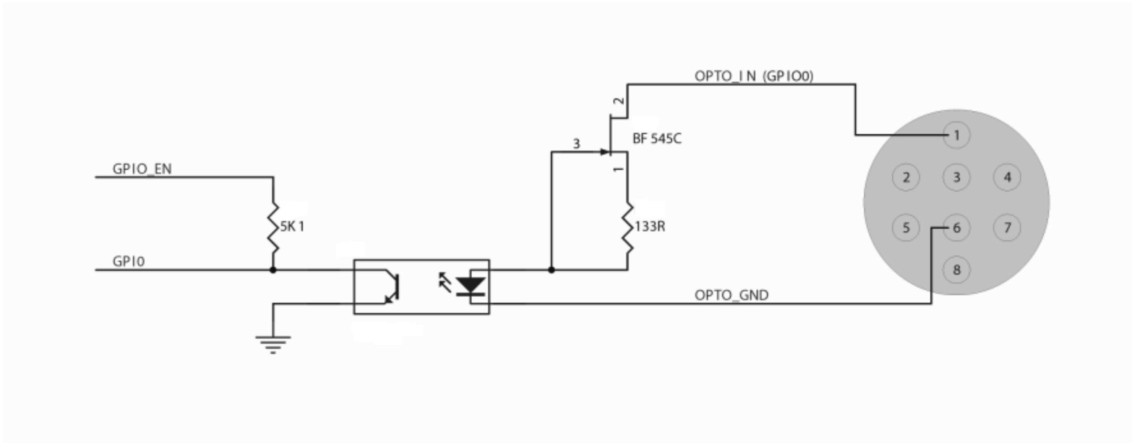


Figure 4.2: Optical input circuit

- Logical 0 input voltage: 0 VDC to +1 VDC (voltage at OPTO_IN)
- Logical 1 input voltage: +1.5 VDC to +30 VDC (voltage at OPTO_IN)
- Maximum input current: 8.3 mA
- Behavior between 1 VDC and 1.5 VDC is undefined and input voltages between those values should be avoided
- Input delay time: 4 µs

4.6.2 GPIO1 (Opto-Isolated Output) Circuit

The figure below shows the schematic for the opto-isolated output circuit. The maximum current allowed through the opto-isolated output circuit is 25 mA.

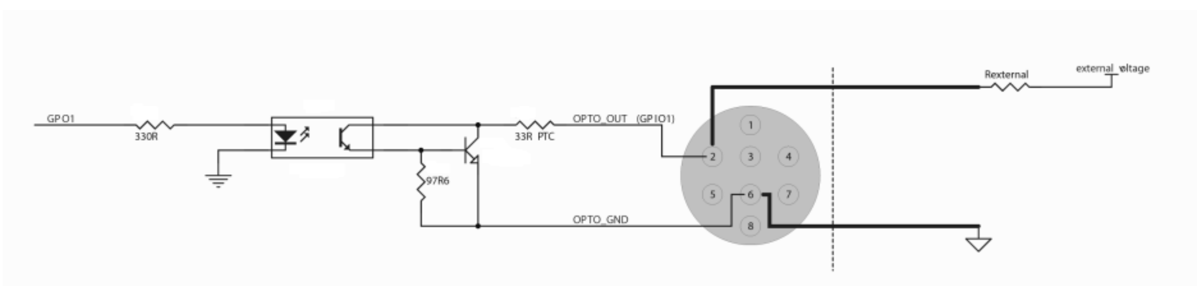


Figure 4.3: Optical output circuit

The following table lists the switching times for the opto-isolator in the output pin, assuming an output VCC of 5 V and a 1 kΩ resistor.

Parameter	Value
Delay Time	9 µs
Rise Time	16.8 µs
Storage Time	0.52 µs
Fall Time	2.92 µs

The following table lists several external voltage and resistor combinations that have been tested to work with the opto-isolated output.

External Voltage	External Resistor	OPTO_OUT Voltage	OPTO_OUT Current
3.3 V	1 kΩ	0.56 V	2.7 mA
5 V	1 kΩ	0.84 V	4.2 mA
12 V	2.4 kΩ	0.91 V	4.6 mA
24 V	4.7 kΩ	1.07 V	5.1 mA

4.6.3 GPIO 2/3 (Bi-Directional) Circuit

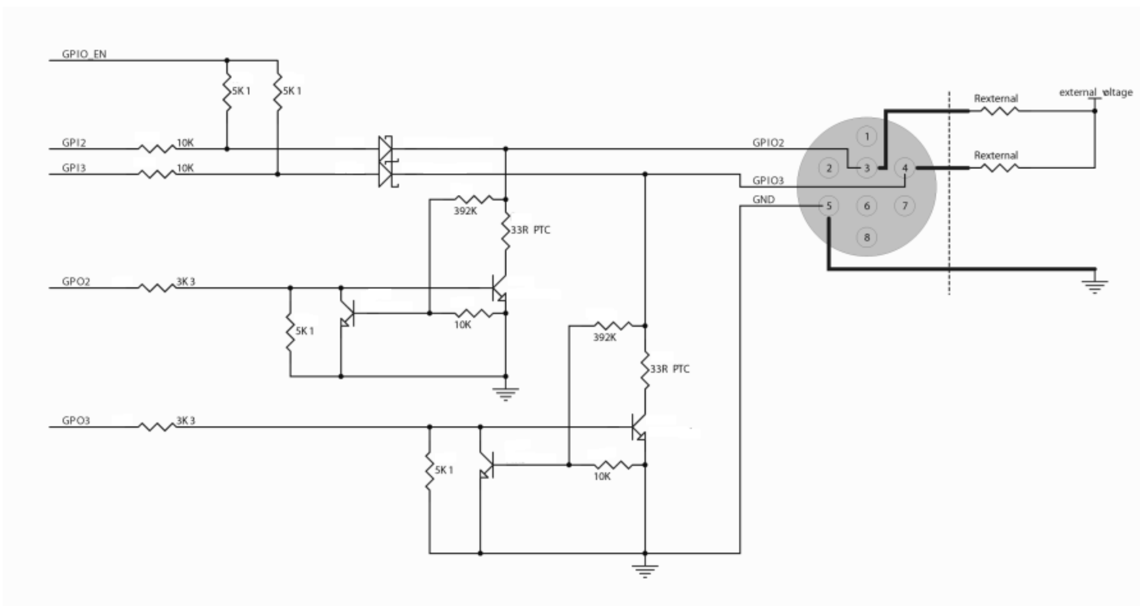


Figure 4.4: GPIO2/3 Circuit

Input Side

- Logical 0 input voltage: 0 VDC to +0.5 VDC (voltage at GPIO2/3)
- Logical 1 input voltage: +1.5 VDC to +30 VDC (voltage at GPIO2/3)
- Behavior between 0.5 VDC and 1.5 VDC is undefined and input voltages between those values should be avoided

To avoid damage, connect the ground (GND) pin first before applying voltage to the GPIO line.

Output Side

The maximum output current allowed through the bi-directional circuit is 25 mA (limit by PTC resistor), and the output impedance is 40 Ω.

The following table lists several external voltage and resistor combinations that have been tested to work with the bi-directional GPIO when configured as output.

External Voltage	External Resistor ($R_{external}$)	GPIO2/3 Voltage
3.3 V	1 kΩ	0.157 V
5 V	1 kΩ	0.218 V
12 V	1 kΩ	0.46 V
24 V	1 kΩ	0.86 V

The following table lists the switching times for a standard GPIO pin, assuming an output VCC of 5V and a 1 kΩ resistor.

Parameter	Value
Delay Time	0.28 µs
Rise Time	0.06 µs
Storage Time	0.03 µs
Fall Time	0.016 µs

5 Video Formats, Modes and Frame Rates

5.1 Frame Rates and Camera Bandwidth

This section is recommended for advanced users only, and is not meant to address all possible applications of the camera

5.1.1 Calculating Maximum Possible Frame Rate

The maximum frame rate allowable for each of the cameras on the bus depends on the resolution of the cameras and the bandwidth, and can be roughly approximated using the following general formula (assuming all cameras are at the same resolution):

Frames_per_second = (Bandwidth / (Pixels_per_frame * Bytes_per_pixel)) / Num_cameras

Example:

To calculate the approximate frames per second available to three 1024x768 cameras in 16-bit mode, you would calculate:

Frames_per_second = (80 MB/s / (1024 * 768 * 2 bytes/pixel)) / 3
= (80 MB/s / 1. 5 MB / frame) / 3
= 53.33 FPS / 3
= 17.8 FPS

The calculation above is only a rough estimate. The IIDC standard defines a specific number of bytes per packet (BPP) for every non-Format 7 video format/mode/frame rate combination. This number is generally higher than the minimum bandwidth that might be expected. In order to accurately determine whether or not there is enough bandwidth available for a given scenario, these numbers must be used. To derive BPP, see Isochronous Bandwidth Requirements: Format 0, Format 1, and Format 2 on page 151.

For example, a single camera in 640x480 RGB mode running at 15 FPS is sending 640 pixels per packet. Each pixel consists of 24 bits, or 3 bytes, of data. Therefore, the camera is sending 640*3 = 1920Bpp of data. The maximum bandwidth of the 1394b bus as discussed above is 8192Bpp, so it would be possible for 8192/1920 = 4 (rounded down) cameras to run in 640x480 RGB mode at 15 FPS on the same 1394b bus.

5.1.2 Maximum Number of Cameras on a Single Bus

- 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 5V, an internal, bus-powered hub provides only 400 mA. Externally-powered hubs provide 500 mA per port.

A single IEEE-1394 OHCI host adapter generally constitutes a single “bus”. There are four elements that limit the number of cameras that can be used on the same 1394 bus:

- Although the 1394b standard limits the maximum number of simultaneous isochronous channels to 16, there is currently no host adapter that is capable of supporting 16 channels.
- The maximum bandwidth of the 1394b bus is 800Mbits/sec (10240Bytes/packet - 8000 cycles/sec). The usable bandwidth as defined by the 1394 Trade Association and enforced by the Microsoft Windows 1394 driver stack (1394bus.sys, ohci1394.sys, etc.) is approximately 80% or 80MBytes/sec (8192 bytes/packet). The remaining 20% of the bandwidth is allocated for asynchronous communication (e.g. register reads/writes). Outside of the Microsoft stack, it may be possible to allocate up to 9830 bytes/packet by using the Point Grey FirePRO driver stack.
- The 1394b standard limits the maximum number of devices on a single bus to 63.
- An inadequate power supply. Consult the voltage and power requirements in the *General Specifications* section to determine the amount of power required to operate the cameras effectively.

Related Knowledge Base Articles

Title	Article
Differences to consider when selecting an IEEE-1394 PCI/PCMCIA host adapter card	Knowledge Base Article 146

5.2 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 camera's ADC (page 14), which digitizes the images , 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.

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

5.2.1 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.

5.2.2 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.

5.2.3 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.

5.2.4 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

5.2.5 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 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.

5.2.5.1 DATA_DEPTH: 630h

This register allows the user to control the endianness of Y16 images.

Format:

Field	Bit	Description
Data_Depth	[0-7]	Effective data depth of current image data. If read value of Data_Depth is zero, shall ignore this field. Read: Effective data depth Write: Ignored
Little_Endian	[8]	Little endian mode for 16-bit pixel formats only Write/Read: 0: Big endian mode (default on initialization) 1: Little endian mode
	[9-31	Reserved

5.2.6 Y8 or Y16 Raw Bayer Output

When operating in Y8 or Y16 mode, color models can output either grayscale or raw Bayer data.

Selecting a half-width, half-height image size and monochrome pixel format, such as 800x600 Y8, using a non-Format 7 mode provides a monochrome binned image. In some cases, enabling raw Bayer output in mono mode provides a raw Bayer region of interest of 800x600, centered within the larger pixel array. This has an effect on the field of view.

5.2.6.1 BAYER_MONO_CTRL: 1050h

This register enables raw Bayer output in non-Format 7 Y8/Y16 modes, or Format 7 Mono8/Mono16 modes.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature. 0: Not Available, 1: Available
	[1-30]	Reserved.
Bayer_Mono_Ctrl	[31]	Value 0: Disable raw Bayer output in mono modes, 1: Enable raw Bayer output in mono modes

5.3 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.

The figures below illustrate how binning works. 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.

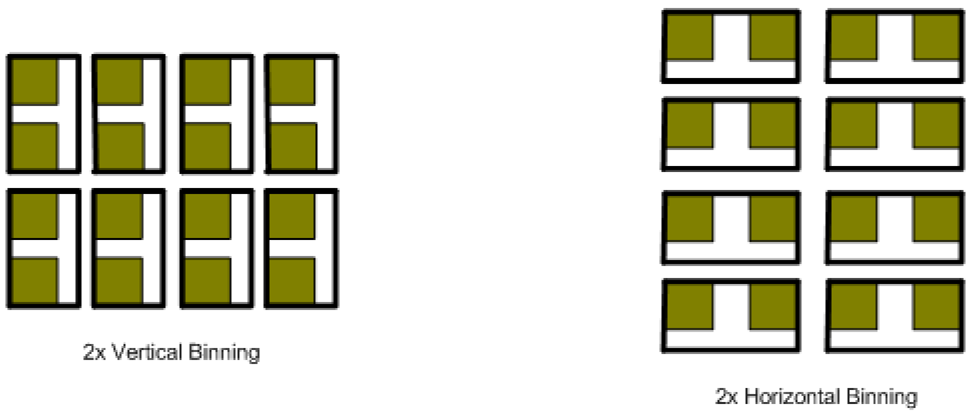


Figure 5.1: 2x Vertical and 2x Horizontal Binning

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.

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 (isochronous transmission disabled) and restarted (iso enabled), unless the change is illegal (e.g. moving the ROI outside the imaging area) or would affect the isochronous packet size.

Additional binning information can be obtained by reading the FORMAT_7_RESIZE_INQ (page 143) register 0x1AC8. 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 *(page 122)* *is not done in any of the binning modes.*

5.3.1 Format 7 Mode Descriptions

Not all camera models support all video modes. For information on which modes are supported by each model, see *Supported Formats, Modes and Frame Rates on next page.*

Mode 0

Mode 0 allows only for specifying a region of interest, and does not perform any binning. This mode uses a relatively fast pixel clock, which can result in reduced smear, and faster frame rates when ROI height is reduced. Overall image quality may vary with frame rate and ROI size/position. Extended shutter times in this mode are limited.

Mode 1

Mode 1 implements a combination of 2X horizontal and 2X vertical additive binning. In color models, all binning is performed after the analog to digital conversion. In monochrome models, vertical binning is performed before the analog to digital conversion, while horizontal binning is performed after the conversion to digital. This mode results in a resolution that is both half the width and half the height of the original image. Mode 1 may result in an increase in brightness and improved signal-to-noise ratio. On monochrome models frame rate will increase; however, on color models, no frame rate increase is achieved.

Mode 4

Mode 4 implements a combination of 2X horizontal and 2X vertical binning. This mode is available on color models only, and produces only monochrome output. Vertical binning is performed prior to color processing, and horizontal binning is performed after color processing. Although image quality may be poorer than in Mode 1, a frame rate increase is possible in this mode.

Mode 5

Mode 5 implements a combination of 4X horizontal and 4X vertical additive binning, resulting in a resolution that is both one quarter the width and one quarter the height of the original image. In color models, all binning is performed after the analog to digital conversion. In monochrome models, vertical binning is performed before the analog to digital conversion, while horizontal binning is performed after the conversion to digital. Mode 5 may result in an increase in brightness and improved signal-to-noise ratio. On monochrome models frame rate will increase; however, on color models, no frame rate increase is achieved.

Mode 6

Mode 6 implements a combination of 4X horizontal and 4X vertical binning. This mode is available on color models only, and produces only monochrome output. Vertical binning is performed prior to color processing, and horizontal binning is performed after color processing. Although image quality may be poorer than in Mode 5, a frame rate increase is possible in this mode.

5.3.2 Exceeding Bandwidth Limitations Using Format 7

There is a mechanism for effectively bypassing bus bandwidth negotiation when using cameras in Format 7 partial image mode. This functionality is useful in any situation where the user is trying to host multiple cameras on the same bus in a configuration that would normally exceed the bandwidth allocation, but where the cameras are configured to transmit data in a manner that does not exceed the total bandwidth. An interactive bandwidth calculator is available in Knowledge Base Article 22. It can be used to calculate approximate bandwidth requirements for various modes.

Related Knowledge Base Articles

Title	Article
Maximum number of IEEE-1394 cameras on a single 1394 bus	Knowledge Base Article 22
Exceeding bandwidth limitations when using Format 7 partial image modes	Knowledge Base Article 256

5.4 Supported Formats, Modes and Frame Rates

The tables on the following pages show the supported pixel formats and mode combinations, along with achievable frame rates at varying resolutions, for each camera model.

5.4.1 GX-FW-10K3 Video Modes

5.4.1.1 GX-FW-10K3 Standard Formats, Modes and Frame Rates

Modes	1.875 FPS	3.75 FPS	7.5 FPS	15 FPS	30 FPS	60 FPS	120 FPS
640x480 Y8	•	•	•	•	•	•	•
640x480 Y16	•	•	•	•	•	•	
1024x768 Y8	•	•	•	•	•	•	
1024x768 Y16	•	•	•	•	•		

5.4.1.2 GX-FW-10K3 Custom Formats, Modes and Frame Rates

Mode	Pixel Format	Max Size (HxV)	Unit Size (H, V)	Maximum Frame Rate			
				Max Size	640x480	320x240	160x120
0	Mono8	1024x1024	8,2	70	112	170	230
0	Mono12	1024x1024	8,2	46	112	170	230
0	Mono16	1024x1024	8,2	35	112	170	230
1	Mono8	512x512	4,2	180		182	230
1	Mono12	512x512	4,2	176		182	230
1	Mono16	512x512	4,2	136		182	230
5	Mono8	256x256	4,2	255			260
5	Mono12	256x256	4,2	255			260
5	Mono16	256x256	4,2	255			260

5.4.2 GX-FW-28S5 Video Modes

5.4.2.1 GX-FW-28S5 Standard Formats, Modes and Frame Rates

Model: • 28S5M • 28S5C						
Modes	**1.875 FPS**	**3.75 FPS**	**7.5 FPS**	**15 FPS**	**30 FPS**	**60 FPS**
800 x 600 Y8			• •	• •	• •	• •
800 x 600 Y16		• •	• •	• •	• •	• •
800 x 600 RGB			•	•	•	
800 x 600 YUV422		•	•	•	•	
1600 x 1200 Y8	• •	• •	• •	• •	• •	
1600 x 1200 Y16	• •	• •	• •	• •		
1600 x 1200 RGB	•	•	•			
1600 x 1200 YUV422	•	•	•	•		
Modes default to highest frame rate						

5.4.2.2 GX-FW-28S5 Custom Formats, Modes and Frame Rates

Images acquired by color cameras using Mono8, Mono12 or Mono16 modes are converted to greyscale on the camera. Users interested in accessing the raw Bayer data to apply their own color conversion algorithm or one of the FlyCapture library algorithms should refer to *Accessing Raw Bayer Data on page 114**.*

Table 5.1: GX-FW-28S5M

Mode	Pixel Format	Maximum Size (H,V)	Unit Size (H,V)	Maximum Frame Rate at:					
				Max Size	1600 x 1200	1280 x 960	640 x 480	320 x 240	160 x 120
0	Mono8	1920x1440	8, 2	26	29	34	50	66	78
0	Mono16	1920x1440	8, 2	14	20	30	50	66	78
0	RAW16	1920x1440	8, 2	14	20	30	50	66	78
0	Mono12	1920x1440	8, 2	18	26	34	50	66	78
1	Mono12	960x720	4, 2	42			50	62	70
1	RAW16	960x720	4, 2	42			50	62	70
1	Mono16	960x720	4, 2	42			50	62	70
1	Mono8	960x720	4, 2	42			50	62	70
5	Mono8	480x360	2, 2	62				62	62
5	Mono16	480x360	2, 2	62				62	62
5	RAW16	480x360	2, 2	62				62	62
5	Mono12	480x360	2, 2	62				62	62

Table 5.2: GX-FW-28S5C

Mode	Pixel Format	Maximum Size (H,V)	Unit Size (H,V)	Maximum Frame Rate at:					
				Max Size	1600 x 1200	1280 x 960	640 x 480	320 x 240	160 x 120
0	Mono8	1920x1440	8, 2	26	29	34	50	66	78
0	Mono12	1920x1440	8, 2	18	26	34	50	66	78
0	Mono16	1920x1440	8, 2	14	20	30	50	66	78
0	RAW8	1920x1440	8, 2	26	29	34	50	66	78
0	RAW12	1920x1440	8, 2	18	26	34	50	66	78
0	RAW16	1920x1440	8, 2	14	20	30	50	66	78
0	YUV411	1920x1440	8, 2	18	26	34	50	66	78
0	YUV422	1920x1440	8, 2	14	20	30	50	66	78
0	YUV444	1920x1440	8, 2	9	12	20	50	66	78
0	RGB8	1920x1440	8, 2	9	12	20	50	66	78
1	Mono8	960x720	4, 2	26			34	50	66
1	Mono12	960x720	4, 2	26			34	50	66
1	Mono16	960x720	4, 2	26			34	50	66
1	RAW8	960x720	4, 2	26			34	50	66
1	RAW12	960x720	4, 2	26			34	50	66
1	RAW16	960x720	4, 2	26			34	50	66
1	YUV411	960x720	4, 2	26			34	50	66
1	YUV422	960x720	4, 2	26			34	50	66
1	YUV444	960x720	4, 2	26			34	50	66
1	RGB8	960x720	4, 2	26			34	50	66
4	Mono8	960x720	4, 2	42			50	62	70
4	Mono12	960x720	4, 2	42			50	62	70
4	Mono16	960x720	4, 2	42			50	62	70
5	Mono8	480x360	2, 2	26				34	50
5	Mono12	480x360	2, 2	26				34	50
5	Mono16	480x360	2, 2	26				34	50
5	YUV411	480x360	2, 2	26				34	50
5	YUV444	480x360	2, 2	26				34	50
5	YUV422	480x360	2, 2	26				34	50
5	RGB8	480x360	2, 2	26				34	50
6	Mono8	480x360	2, 2	62				62	62
6	Mono12	480x360	2, 2	62				62	62
6	Mono16	480x360	2, 2	62				62	62

5.4.3 GX-FW-60S6 Video Modes

5.4.3.1 GX-FW-60S6 Standard Formats, Modes and Frame Rates

Model: • 60S6M • 60S6C						
Modes	**1.875 FPS**	**3.75 FPS**	**7.5 FPS**	**15 FPS**	**30 FPS**	**60 FPS**
640x 480 Y8	• •	• •	• •	• •	• •	• •
640 x 480 Y16	• •	• •	• •	• •	• •	• •
1280 x 960 Y8	• •	• •	• •	• •	• •	
1280 x 960 Y16	• •	• •	• •	• •		
1280 x 960 RGB	•	•	•	•		
1280 x 960 YUV422	•	•	•	•		
1600 x 1200 Y8	• •	• •	• •	• •	• •	
1600 x 1200 Y16	• •	• •	• •	• •		
1600 x 1200 RGB	•	•	•			
1600 x 1200 YUV422	•	•	•	•		
Modes default to highest frame rate						

5.4.3.2 GX-FW-60S6 Custom Formats, Modes and Frame Rates

Images acquired by color cameras using Mono8, Mono12 or Mono16 modes are converted to greyscale on the camera. Users interested in accessing the raw Bayer data to apply their own color conversion algorithm or one of the FlyCapture library algorithms should refer to Accessing Raw Bayer Data on page 114.

Table 5.3: GX-FW-60S6M

Mode	Pixel Format	Maximum Size	Maximum Frame Rate at:					
			Maximum Size	1600 x 1200	1280 x 960	640 x 480	320 x 240	160 x 120
0	Mono8	2736x2192	11.5	20	22	31	38	44
0	Mono16	2736x2192	6	19	22	31	38	44
0	Raw16	2736x2192	6	19	22	31	38	44
0	Mono12	2736x2192	8	20	22	31	38	44
1	Mono12	1368x1096	23		24	31	36	40
1	Raw16	1368x1096	23		24	31	36	40
1	Mono16	1368x1096	23		24	31	36	40
1	Mono8	1368x1096	23		24	31	36	40
5	Mono8	684x548	36			36	36	36
5	Mono16	684x548	36			36	36	36
5	Mono12	684x548	36			36	36	36

Table 5.4: GX-FW-60S6C

Mode	Pixel Format	Maximum Size	Maximum Frame Rate at:					
			Maximum Size	1600 x 1200	1280 x 960	640 x 480	320 x 240	160 x 120
0	Mono8	2736x2192	11.5	20	22	31	38	44
0	Raw12	2736x2192	8	20	22	31	38	44
0	Mono12	2736x2192	8	20	22	31	38	44
0	RGB8	2736x2192	3.5	13	20	31	38	44
0	444YUV8	2736x2192	3.5	13	20	31	38	44
0	422YUV8	2736x2192	6	19	22	31	38	44
0	411YUV8	2736x2192	8	20	22	31	38	44
0	Raw16	2736x2192	6	19	22	31	38	44
0	Raw8	2736x2192	11.5	20	22	31	38	44
0	Mono16	2736x2192	6	19	22	31	38	44
1	Raw12	1368x1096	13		14	22	31	38
1	Mono12	1368x1096	13		14	22	31	38
1	RGB8	1368x1096	13		14	22	31	38
1	444YUV8	1368x1096	13		14	22	31	38
1	422YUV8	1368x1096	13		14	22	31	38
1	411YUV8	1368x1096	13		14	22	31	38
1	Raw16	1368x1096	13		14	22	31	38
1	Raw8	1368x1096	13		14	22	31	38
1	Mono16	1368x1096	13		14	22	31	38
1	Mono8	1368x1096	13		14	22	31	38
4	Mono12	1368x1096	23		24	31	36	40
4	Mono16	1368x1096	23		24	31	36	40
4	Mono8	1368x1096	23		24	31	36	40
5	Mono12	684x548	13			14	22	31
5	RGB8	684x548	13			14	22	31
5	444YUV8	684x548	13			14	22	31
5	422YUV8	684x548	13			14	22	31
5	411YUV8	684x548	13			14	22	31
5	Mono16	684x548	13			14	22	31
5	Mono8	684x548	13			14	22	31
6	Mono8	684x548	36			36	36	36
6	Mono16	684x548	36			36	36	36
6	Mono12	684x548	36			36	36	36

5.5 Video Format, Mode, and Frame Rate Settings

The following settings control the video format and mode of the camera.

Frame Rate—This provides control over the frame rate of the camera. When this feature is in auto mode, exposure time is limited by the frame rate value dynamically, which is determined by the Current Frame Rate. When this feature is in manual mode, the actual frame interval (time between individual image acquisitions) is fixed by the frame rate value. The available frame rate range depends on the current video format and/or video mode.

This is set to OFF when the camera is operating in asynchronous trigger mode. For more information, see Asynchronous Triggering on page 77.

Current Frame Rate—Allows the user to query and modify the current frame rate of the camera.

Current Video Mode—Allows the user to query and modify the current video mode of the camera.

Current Video Format—Allows the user to query and modify the current video format of the camera.

5.5.1 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 147)*.

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

Field	Bit	Description
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.

Related Resources

Title	Link
FlyCapture SDK *ExtendedShutterEx* sample program	ExtendedShutterEx

5.5.2 CURRENT_FRAME_RATE: 600h

Format:

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

5.5.3 CURRENT_VIDEO_MODE: 604h

Format:

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

5.5.4 CURRENT_VIDEO_FORMAT: 608h

Format:

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

5.5.5 Example: Setting a Standard Video Mode, Format and Frame Rate Using the FlyCapture API

The following FlyCapture2 code snippet sets the camera to: 640x480 Y8 at 60 FPS.

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

6 Image Acquisition

6.1 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 the isochronous packet section of the Appendix (page 151).

6.1.1 Camera Power and Isochronous Transmission

If isochronous transmission 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 will revert to its default factory settings, or if applicable, the last saved memory channel. For more information, see User Memory Channels on page 28.

6.1.2 When Camera Property Settings Take Effect

When the camera is in isochronous (free-running) transmission mode, it is not possible to guarantee that camera setting changes are applied to the 'next' image (that is, the image that is grabbed after a new property value is written). Properties such as gain and white balance are controlled by the analog-to-digital (A/D) converter, which makes it difficult to determine the exact amount of time required for certain settings to take effect.

In general, properties are applied to either the next image (n) or the one after next (n+1). Specifically:

- If the camera is in asynchronous (trigger) mode and gain is set before digitization (i.e. before analog data reaches the A/D converter and is sent out along the FireWire to the PC), the setting is applied to the next image. Using the camera in asynchronous (trigger) mode is the most reliable way of ensuring that camera property changes are applied to the next frame.
- If the camera is in isochronous (free-running) mode and gain is set, there is a low probability that the gain will be applied to the next image (n). In most cases, the gain setting is applied to the image after next (n+1). This is because the shutter period may be quite short (depending on the camera frame rate), which may not provide enough time for the setting to be applied before the data off the CCD reaches the A/D converter. CCD cameras tend to act similarly in terms of providing full control to the shutter. CCD cameras allow you to set the shutter at any time, and apply it to the next image or the one after next. (See Shutter on page 91 for more

information.) The amount of time for properties to become effective is also correlated with the frame rate. The higher the frame rate, the less certainty of when the effect takes place.

When the camera is running in continuous shot (free-running) mode, the worst-case scenario would require up to four (4) frames for a setting to take effect. The following list enumerates the sources of delay just after the parameter is set:

- All images that are currently buffered on the PC. These images are not affected by the latest parameter settings, but still must be grabbed. This is at most 1 image, except in cases where a variation of the BUFFER_ FRAMES enumerator is used (for users of the FlyCapture2 SDK). In this case, the number of buffers that have been allocated minus 1 images needs to be grabbed.
- The image that is currently being transmitted from the camera. This image is not affected by the latest parameter setting, but must still be grabbed. In free-running mode, images are effectively being transmitted all the time.
- The image that is currently being integrated. It is possible that the parameter setting happened too late to affect the current image.

For example, consider 4 frames: A, B, C, D.

1. Frame A is already captured and waiting on the PC.
2. Frame B is just being captured.
3. The property value (ie: shutter, gain, etc.) is changed (while Frame B is being captured).
4. Frame C is captured. The new property value is not applied due to latency, type of property, and other factors).
5. Frame D is captured with the new property value.

To be sure you are obtaining an image with the new property applied to it, it is safest to take the 4th frame. In cases where it is important to determine and react to parameter changes in a faster manner, check the embedded image information (page 116) for the parameters applied to a specific frame. Note, however, that this information reads property values that have been written by the end of shutter integration, even if they are not yet in effect.

6.1.3 ISO_CHANNEL/ISO_SPEED: 60Ch

Allows the user to query the camera's isochronous transmission channel and speed information.

Format:

Field	Bit	Description
ISO_Channel	[0-3]	Isochronous channel number for video data transmission (Except for Format_6)
	[4-5]	Reserved
ISO_Speed	[6-7]	Isochronous transmit speed code. (Except for Format_6) 0 = 100 Mbps 1 = 200 Mbps 2 = 400 Mbps
	[8-15]	Reserved

Field	Bit	Description
Operation_Mode	[16]	1394 operation mode Change control register sets of ISO_Channel and ISO_Speed registers 0 = Legacy (v1.30 compatible), 1 = 1394.b (v1.31 mode) Camera shall start in legacy mode for backward compatibility
	[17]	Reserved
ISO_Channel_B	[18-23]	Isochronous channel number for video data transmission of 1394.b mode (Except for Format_6)
	[24-28]	Reserved
ISO_Speed_B	[29-31]	Isochronous transmit speed code of 1394.b mode. (Except for Format_6) 0 = 100 Mbps 1 = 200 Mbps 2 = 400 Mbps 3 = 800 Mbps 4 = 1.6 Gbps 5 = 3.2 Gbpss

6.1.4 ISO_EN/CONTINUOUS_SHOT: 614h

This register allows the control of isochronous data transmission. During ISO_EN = 1 or One_Shot = 1 or Multi_Shot =1, the register value which reflects the Isochronous packet format cannot change. Data transfer control priority is ISO_EN > One_Shot > Multi_Shot.

Format:

Field	Bit	Description
ISO_EN/Continuous Shot	[0]	0 = Stop ISO transmission of video data. Continuous Shot is not enabled. 1 = Start ISO transmission of video data.
	[1-31]	Reserved.

6.1.5 ONE_SHOT/MULTI_SHOT: 61Ch

This register allows the user to control single and multi-shot functionality of the camera. During ISO_EN = 1, *One_Shot* = 1 or *Multi_Shot* =1, the register value which reflects the Isochronous packet format cannot change. Data transfer control priority is ISO_EN > One_Shot > Multi_Shot.

Format:

Field	Bit	Description
One_Shot	[0]	1 = only one frame of video data is transmitted. (Self cleared after transmission) Ignored if ISO_EN = 1
Multi_Shot	[1]	1 = N frames of video data is transmitted. (Self cleared after transmission) Ignored if ISO_EN = 1 or One_Shot =1
	[2-15]	Reserved.
Count_Number	[16-31]	Count number for Multi-shot function.

6.2 Automatic Inter-Camera Synchronization

Multiple Point Grey cameras, when they are on the same bus and running at the same frame rate, are automatically synchronized to each other at the hardware level. When using multiple cameras, the timing of one camera to another camera is as follows:

- If the cameras are on the same bus, the cameras are synchronized to within 125 µs (microseconds) of each other (note: 125 µs is the maximum deviation).
- If the cameras are on separate buses, use PointGrey's MultiSync™ software to synchronize the cameras across buses. This can be used to synchronize cameras on different buses within the same computer or on different buses across multiple computers. The software will ensure that the cameras are synchronized to within 125 µs. If Multisync is not running, there is no timing correlation between separate cameras on separate buses.

It is possible to offset the synchronization of individual cameras relative to other cameras using the TRIGGER_DELAY register 0x834 (page 87).

6.3 Asynchronous Triggering

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

Grasshopper Express Supported Trigger Modes			
Model	Mode		More Information
All	0	Standard	page 79
All	1	Bulb	page 80
All	3	Skip Frames	page 80
All	14	Overlapped	page 81
All	15	Multishot	page 81

6.3.1 External Trigger Timing

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

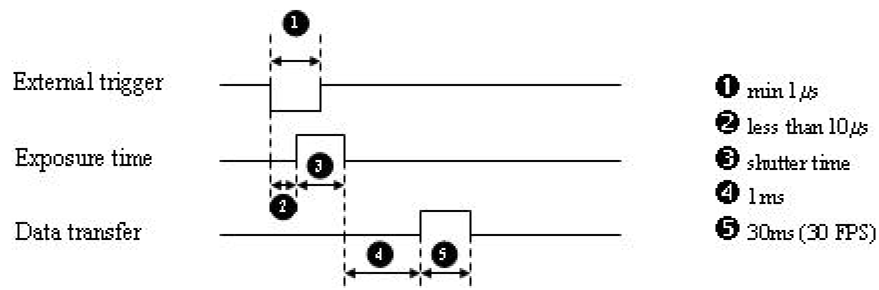


Figure 6.1: External trigger timing characteristics

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 on page 38) 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.

6.3.2 Minimum Trigger Pulse Length

A digital signal debouncer helps to ensure that the camera does not respond to spurious electrical signals that are shorter than 16 ticks of the current pixel clock setting. This safeguard results in a minimum 16-tick delay before the camera responds to a trigger signal. The pixel clock frequency can be read from the floating point PIXEL_CLOCK_FREQ register 0x1AF0 (page 27).

6.3.3 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.

6.3.4 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.

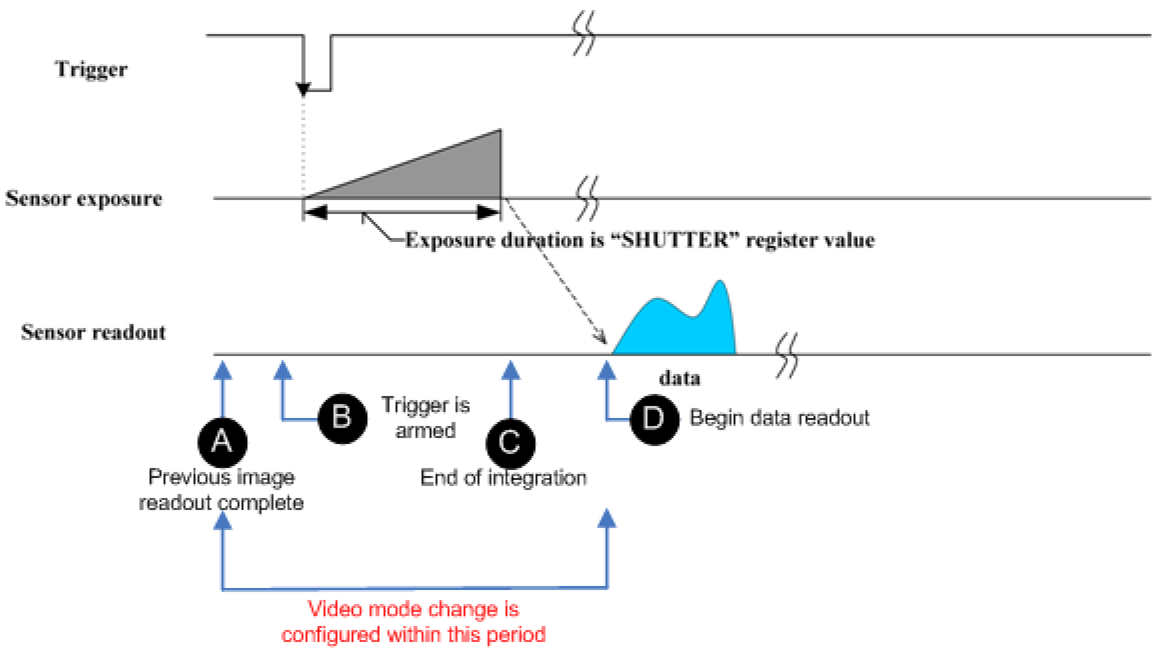


Figure 6.2: Relationship Between External Triggering and Video Mode Change Request

When operating in trigger mode 0 (page 79) or trigger mode 1 (page 79), 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 trigger mode 14 (page 81), point B occurs before point A. The result is that, in most cases, there is a delay of one triggered image for a video mode request, made before the configuration period, to take effect. In trigger mode 15 (page 81), change requests made after point A for any given image readout are honored only after a delay of one image.

6.3.5 Trigger Modes

6.3.5.1 Trigger Mode 0 ("Standard External Trigger Mode")

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 valuedescribes 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 Trigger Mode 14.

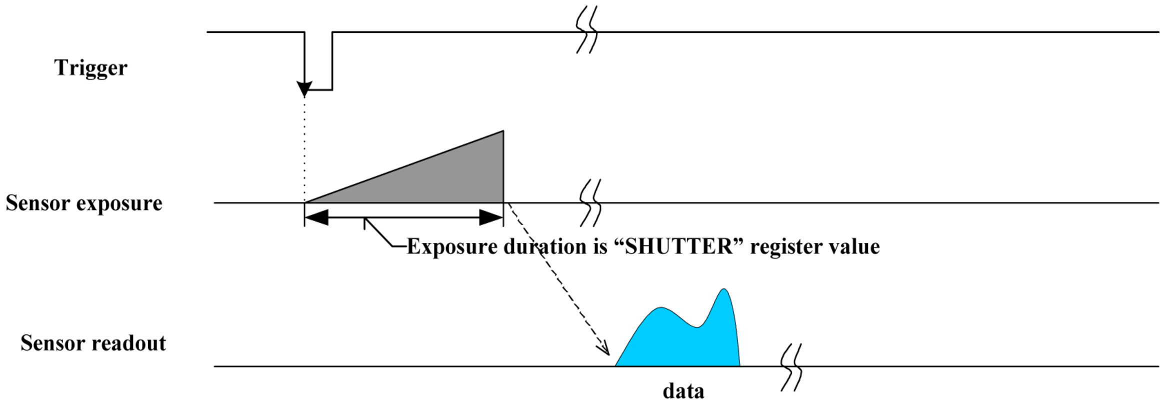


Figure 6.3: Trigger Mode 0 ("Standard External Trigger Mode")

6.3.5.2 Trigger Mode 1 ("Bulb Shutter Mode")

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.

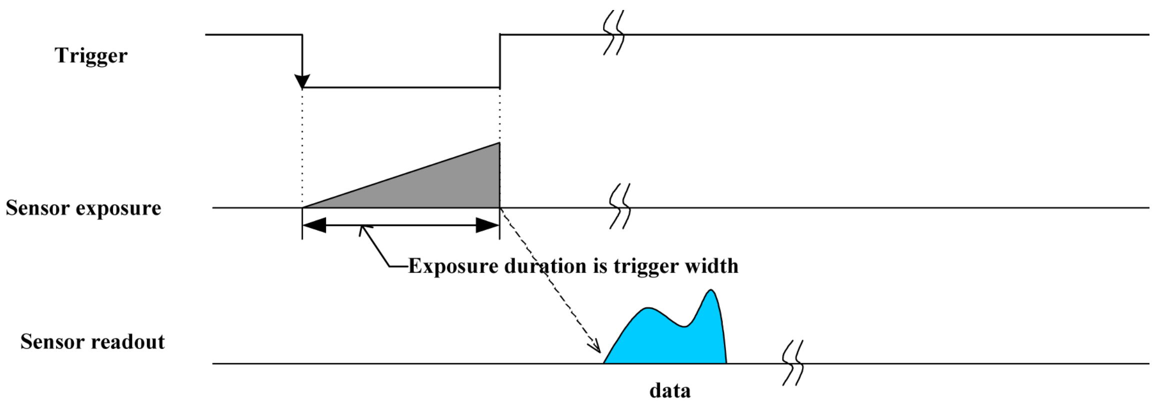


Figure 6.4: Trigger Mode 1 ("Bulb Shutter Mode")

6.3.5.3 Trigger Mode 3 ("Skip Frames Mode")

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. Note that this is different from the IIDC specification that states the cycle time will be N times greater than the fastest frame rate.

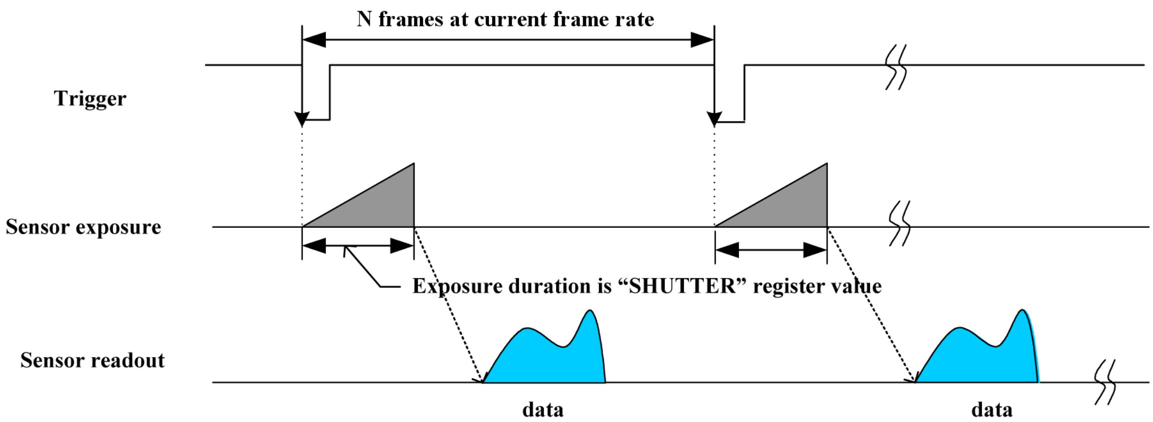


Figure 6.5: Trigger Mode 3 ("Skip Frames Mode")

6.3.5.4 Trigger Mode 14 ("Overlapped Exposure/Readout Mode")

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.

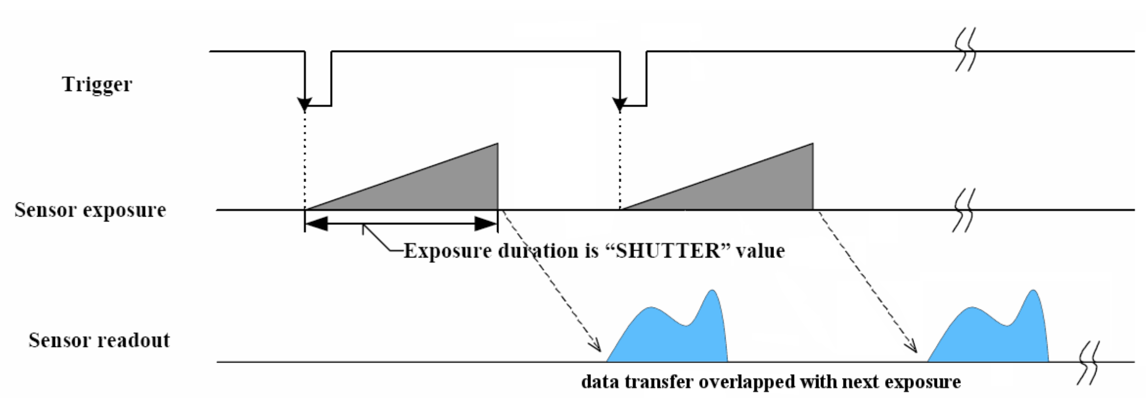


Figure 6.6: Trigger Mode 14 ("Overlapped Exposure/Readout Mode")

6.3.5.5 Trigger Mode 15 ("Multi-Shot Trigger Mode")

Trigger Mode 15 is a vendor-unique trigger mode that allows the user to fire a single hardware or software trigger and have the camera acquire and stream a predetermined number of images at the current frame rate.

The number of images to be acquired is determined by the Parameter field of the TRIGGER_MODE register 0x830 (page 86), which allows up to 255 images to be acquired from a single trigger. Writing a value of 0 to the parameter field will result in an infinite number of images to be acquired, essentially allowing users to trigger the camera into a free-running mode. Once the trigger is fired, the camera will acquire N images with an exposure time equal to the value defined by the SHUTTER register, and stream the images to the host system at the current frame rate. Once this is complete, the camera can be triggered again to repeat the sequence.

Any write to the TRIGGER_MODE register 0x830 will cause the current sequence to stop.

During the capture of N images, the camera is still in an asynchronous trigger mode (essentially Trigger Mode 14), rather than continuous (free-running) mode. The result of this is that the FRAME_RATE register 0x83C will be turned OFF, and the camera put into extended shutter mode. Users should therefore ensure that the maximum shutter time is limited to 1/frame_rate to get the N images captured at the current frame rate.

Related Knowledge Base Articles

Title	Article
Extended shutter mode operation for DCAM-compliant PGR Imaging Products	Knowledge Base Article 166

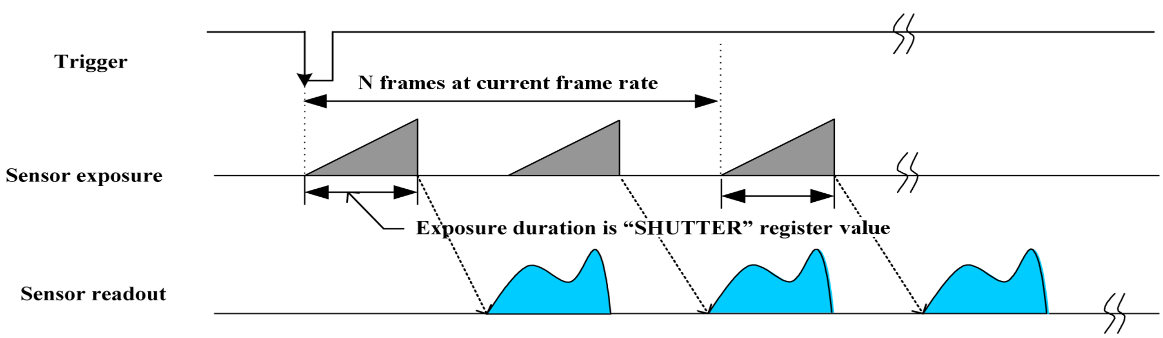


Figure 6.7: Trigger Mode 15 ("Multi-Shot Trigger Mode")

6.3.6 Example: Asynchronous Hardware Triggering (Using the Camera Registers)

The following example illustrates how to synchronize image acquisition to a trigger from an external hardware device in Trigger Mode 0 (page 79).

Determine the Default External Trigger Pin

One of the camera GPIO pins is configured as the default trigger. To determine which pin is the default input/trigger pin either:

1. See *General Purpose Input/Output*; or
2. Get the value of the TRIGGER_MODE register 0x830 (page 86). The *Trigger_Source* field (bits 8-10) is the current trigger source. For example, if the value represented by the *Trigger_Source* field is 0, the default trigger source is GPIO0.

For example:

0x830 = 0x80100000

8	0	1	0	0	0	0	0	Hex
1000	0000	0001	0000	0000	0000	0000	0000	Binary
0-7		8-15		16-23		24-31		Bits

This indicates that a Trigger Mode is available (bit 0 = 1) but not currently enabled (bit 6 = 0). It also indicates that GPIO0 is the default trigger pin (bits 8-10 = 0), and the default polarity of the pin is active low (bit 7 = 0), which means the camera will trigger on the falling edge of a pulse.

Configure a Different GPIO Pin to be an External Trigger

If you wish to use a different GPIO pin as the external trigger instead of the default trigger, you will need to configure the specific pin to be an input trigger, then configure the camera to use this newly allocated trigger pin.

For example, to configure the camera to use GPIO2 as the external trigger pin:

1. Get the value of the PIO_DIRECTION register 0x11F8 (page 87) to determine the current states of each GPIO pin. For example:

0x11F8 = 0x20000000

2	0	0	0	0	0	0	0	Hex
0010	0000	0000	0000	0000	0000	0000	0000	Binary
0-7		8-15		16-23		24-31		Bits

Each of the first four bits represents the current state of its associated GPIO pin: ‘0’ indicates it is a input/trigger, and ‘1’ indicates it is an output/strobe. In the example above, 0x2 = 0010 in binary, so GPIO0, GPIO1 and GPIO 3 are all configured as inputs and GPIO2 is an output.

2. To set GPIO2 in the example above to be an input/trigger, and all other GPIO pins as outputs:

0x11F8 = 0xD0000000

D	0	0	0	0	0	0	0	Hex
1101	0000	0000	0000	0000	0000	0000	0000	Binary
0-7		8-15		16-23		24-31		Bits

3. Configure the camera to use GPIO2 as the external trigger source by setting bits 8-10 of the TRIGGER_MODE register (page 86). For example, for GPIO pin “2”, we set bits 8-10 to 010, which is 2 in binary):

0x830 = 0x8040000000 (assumes bits 11-31 are zero)

8	0	4	0	0	0	0	0	Hex
1000	0000	0100	0000	0000	0000	0000	0000	Binary
0-7		8-15		16-23		24-31		Bits

Enable Trigger Mode

The camera must be put into Trigger Mode 0 to allow it to be externally triggered.

To do this in the FlyCap graphical user interface:

1. Open the Camera Control Dialog
2. Select the “Trigger” tab
3. Check the "Enable/disable trigger" (“Trigger On/Off” in earlier versions) checkbox

To do this by directly accessing the camera’s TRIGGER_MODE register (page 86):

1. Get register 0x830
2. Turn trigger Mode 0 ON by setting bit 6 to one (1) and setting bits 12-15 to zero (0)

Ensuring Trigger is Armed

It is possible for the camera to be in asynchronous trigger mode but not be ready to accept a trigger. The reason is the camera may be currently exposing an image; the camera is only ready to be triggered again when this image finishes integrating.

To ensure that the camera is ready to be triggered, poll the SOFTWARE_TRIGGER register 0x62C (page 87). The concept of polling to ensure the trigger is armed is demonstrated in the AsyncTriggerEx example program distributed with the FlyCapture SDK.

Once the trigger is reporting that it is armed, there should be no delay between when the user can enable isochronous transmission and when they can trigger the camera. In fact, it is possible to trigger the camera before iso is enabled and receive the image that was triggered, provided iso is enabled at some point during exposure. For example, assuming a 10 ms shutter time, it is possible to trigger the camera, enable iso 5 ms after, and still receive the triggered image.

Externally Trigger the Camera

At this point, one of the camera’s GPIO pins should be configured as the external trigger source, the camera should be in Trigger Mode 0, and the trigger is armed and ready to be fired. To acquire an image, connect the external 5V or 3.3V TTL synchronization signal to the GPIO pin. Once the trigger signal is received, an image will be grabbed.

6.3.7 Example: Asynchronous Hardware Triggering (Using the FlyCapture API)

The following FlyCapture 2.x 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);
```

6.3.8 Asynchronous Software Triggering

Shutter integration can be initiated by a register write (software trigger) via SOFTWARE_TRIGGER register 0x62C (page 87).

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

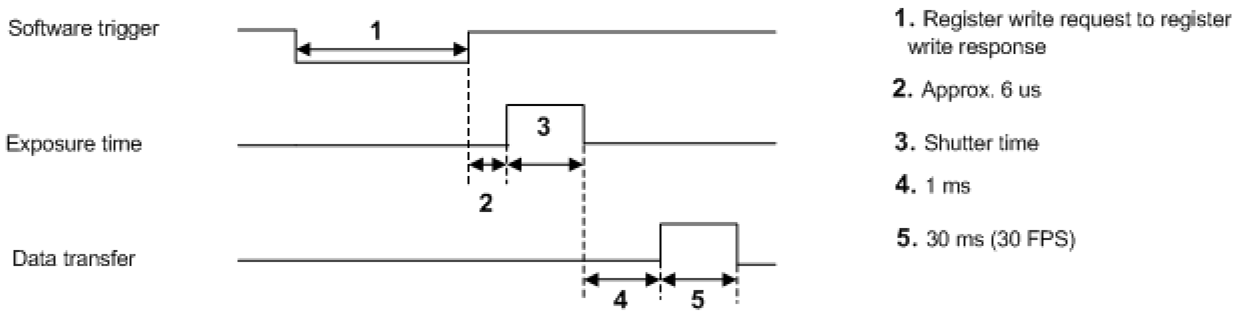


Figure 6.8: Software trigger timing

The time from when the software trigger is written on the camera to when the start of integration occurs can only be approximated. The average time from register write request to register write response is approximately 49.85 us. The "write success" response is only sent from the camera to the host system once the internal trigger pulse is initiated. We then add the trigger latency (time from the trigger pulse to the start of integration) to this, which is approximately 6 us for a camera capturing 640x480 images. Therefore, the total time from when the register is written to the start of integration is approximately 56 us.

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.

6.3.9 Asynchronous Trigger Settings

For information about working with the trigger registers in your FlyCapture application, refer to the AsyncTriggerEx sample 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 91). 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).

- In Free-running mode: controls the synchronization offset of the camera relative to normal synchronization. This is useful for offsetting image acquisition between automatically synchronized cameras. (Not applicable to GigE Vision or USB cameras.)

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

6.3.9.1 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 141).
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 Trigger Modes on page 79. 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)

6.3.9.2 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.

6.3.9.3 PIO_DIRECTION: 11F8h

If the *IOx_Mode* bit is asserted (write a '1'), this means the GPIO pin is currently configured as an output and the *Pin_ Mode* of the GPIO pin (see the GPIO_CTRL_PIN_x register) is GPIO_Mode_8. Otherwise, the *Pin_ Mode* will be GPIO_ Mode_0 (Input). The PIO_DIRECTION register is writeable only when the current GPIO_Mode is GPIO_Mode_0 or GPIO_Mode_8.

Format

Field	Bit	Description
IO0_Mode	[0]	Current mode of GPIO Pin 0 0: Other, 1: Output
IO1_Mode	[1]	Current mode of GPIO Pin 1 0: Other, 1: Output
IO2_Mode	[2]	Current mode of GPIO Pin 2 0: Other, 1: Output
IO3_Mode	[3]	Current mode of GPIO Pin 3 0: Other, 1: Output
	[4-31]	Reserved

6.3.9.4 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

7 Imaging Parameters and Control

7.1 Overview of Imaging Parameters

The camera supports control over the following imaging parameters:

Imaging Parameter	More Information	Register Control	FlyCapture API Examples
Brightness	page 90	BRIGHTNESS: 800h on next page	Example: Setting Brightness Using the FlyCapture API on page 91
Gain	page 94	GAIN: 820h on page 95	Example: Setting Gain Using the FlyCapture API on page 96
Saturation (color models only)	page 106	SATURATION: 814h on page 106	Example: Setting Saturation Using the FlyCapture API on page 107
Hue (color models only)	page 107	HUE: 810h on page 108	
Sharpness	page 109	SHARPNESS: 808h on page 109	Example: Setting Sharpness Using the FlyCapture API on page 110
Gamma	page 100	GAMMA: 818h on page 102	Example: Setting Gamma Using the FlyCapture API on page 105
Lookup Table	page 100	LUT: 80000h – 80048h (IIDC 1.32) on page 103	
White Balance (color models only)	page 111	WHITE_BALANCE: 80Ch on page 112	Example: Setting White Balance Using the FlyCapture API on page 112
Shutter	page 91	SHUTTER: 81Ch on page 93	
Bayer Color Processing (color models only)	page 113	BAYER_TILE_MAPPING: 1040h on page 114	Example: Accessing Raw Bayer Data using FlyCapture2 on page 115
Image Mirror/Flip	page 115	MIRROR_IMAGE_CTRL: 1054h on page 115	
Auto Exposure	page 96	AUTO_EXPOSURE: 804h on page 97	
Embedded Image Information	page 116	FRAME_INFO: 12F8h on page 118	

Most of 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.

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

For information about timing changes to the imaging settings, see When Camera Property Settings Take Effect on page 74.

7.2 Brightness

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

The camera supports brightness control.

7.2.1 BRIGHTNESS: 800h

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 147).

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

Field	Bit	Description
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.

7.2.2 Example: Setting Brightness Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.3 Shutter

The camera 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. 66ms to 133ms).

The supported shutter time range is 0.04 ms to >1.5 seconds (extended shutter mode).

The terms "integration" and "exposure" are often used interchangeably with "shutter".

The time between the end of shutter for consecutive frames will always be constant. However, if the shutter time is continually changing (e.g. shutter is in Auto mode 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 async 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 isochronous data transfer is disabled and images are not being streamed to the computer. The camera continues exposing images even when ISO is off in order to keep things such as the auto exposure algorithm (if enabled) running. This is done to ensure that when a user starts requesting images (ISO turned on), 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.

For more information about the timing of property settings, see When Camera Property Settings Take Effect on page 74.

7.3.1 Extended Shutter Times

The maximum shutter time can be extended beyond the normal shutter range by turning off the frame rate setting. Once the frame rate is turned off, you should see the maximum value of the shutter time increase.

The maximum extended shutter time reported by the SHUTTER_INQ register 51Ch (page 139) is capped at 4095 (0xFFF), the maximum value allowed by the Max_Value field of this register. Use the Max_Value of the ABS_VAL_SHUTTER register (page 147) to determine the maximum shutter.

Related Knowledge Base Articles

Title	Article
Extended shutter mode operation for DCAM-compliant PGR Imaging Products.	Knowledge Base Article 166

Related Resources

Title	Link
FlyCapture SDK *ExtendedShutterEx* sample program	ExtendedShutterEx

7.3.2 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 147). 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 96). Both the *Value* field and the ABS__VAL_SHUTTER register become read only.

The fixed-point (relative) values reported by this register can be converted to absolute values based on the following chart:

Fixed-point Value Range	Equivalent Absolute Value Unit	Equivalent Absolute Value Range
1 to 1024	10 us	0.01 ms to 10.24 ms
1025 to 1536	20 us	10.26 ms to 20.48 ms
1537 to 2048	40 us	20.52 to 40.96 ms
2049 to 2560	80 us	41.04 ms to 81.92 ms
...	...	...

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.

7.3.3 Example: Setting Shutter Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.4 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 camera 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 up to 24 dB in steps of 0.046 dB.

Increasing gain also increases image noise, which can affect image quality. To increase image intensity, try adjusting the lens aperture (iris) and shutter time *(page 91)* *first.*

7.4.1 GAIN: 820h

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 147).

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.

7.4.2 Example: Setting Gain Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.5 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.
ON 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.
ON 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.

7.5.1 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 147).

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
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic

Field	Bit	Description
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.

7.5.2 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

7.5.3 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 register 81Ch *(page 93)* *for the list of ranges).*

7.5.4 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

7.5.5 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

Offset	Name	Field	Bit	Description
Base + Ch	AE_ROI_SIZE	Width	[0-15]	Width of ROI
		Height	[16-31]	Height of ROI

7.5.6 Example: Setting Auto Exposure Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.6 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 on and has a value of 1.25. To obtain a linear response, turn gamma off.

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.

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 up to 16 banks where each bank can contain up to 16 channels. Each channel shall define a table with a length of $2^{\text{Input\_Depth}}$ entries where each entry is *Output_ Depth* bits wide. Channel table entries shall be 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 shall be set by writing to the *Active_Bank* field of the LUT_Ctrl register.

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

Lookup Table Data Structure

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

Cn: Channel Number

En : Entry Number

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

Related Knowledge Base Articles

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

7.6.1 GAMMA: 818h

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 147)*.

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

Field	Bit	Description
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

7.6.2 LUT: 80000h – 80048h (IIDC 1.32)

Offset	Name	Field	Bit	Description
80000h	LUT_Ctrl_Inq (Read Only)	Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
			[1-4]	Reserved
		ON_OFF_Inq	[5]	Capability of turning this feature ON or OFF.
			[6-7]	Reserved
		Input_Depth	[8-12]	Input data bit depth
		Output_Depth	[13-17]	Output data bit depth
			[18]	Reserved
		Number_of_Channels	[19-23]	Number of channels
			[24-26]	Reserved
		Number_of_Banks	[27-31]	Number of banks
80004h	LUT_Bank_Rd_Inq	Read_Bank_0_Inq	[0]	Capability of reading data from Bank 0
		Read_Bank_1_Inq	[1]	Capability of reading data from Bank 1
		Read_Bank_2_Inq	[2]	Capability of reading data from Bank 2
		Read_Bank_3_Inq	[3]	Capability of reading data from Bank 3
		Read_Bank_4_Inq	[4]	Capability of reading data from Bank 4
		Read_Bank_5_Inq	[5]	Capability of reading data from Bank 5
		Read_Bank_6_Inq	[6]	Capability of reading data from Bank 6
		Read_Bank_7_Inq	[7]	Capability of reading data from Bank 7
		Read_Bank_8_Inq	[8]	Capability of reading data from Bank 8
		Read_Bank_9_Inq	[9]	Capability of reading data from Bank 9
		Read_Bank_10_Inq	[10]	Capability of reading data from Bank 10
		Read_Bank_11_Inq	[11]	Capability of reading data from Bank 11
		Read_Bank_12_Inq	[12]	Capability of reading data from Bank 12
		Read_Bank_13_Inq	[13]	Capability of reading data from Bank 13
		Read_Bank_14_Inq	[14]	Capability of reading data from Bank 14
		Read_Bank_15_Inq	[15]	Capability of reading data from Bank 15

Offset	Name	Field	Bit	Description
	LUT_Bank_Wr_Inq	Write_Bank_0_Inq	[16]	Capability of writing data to Bank 0
		Write_Bank_1_Inq	[17]	Capability of writing data to Bank 1
		Write_Bank_2_Inq	[18]	Capability of writing data to Bank 2
		Write_Bank_3_Inq	[19]	Capability of writing data to Bank 3
		Write_Bank_4_Inq	[20]	Capability of writing data to Bank 4
		Write_Bank_5_Inq	[21]	Capability of writing data to Bank 5
		Write_Bank_6_Inq	[22]	Capability of writing data to Bank 6
		Write_Bank_7_Inq	[23]	Capability of writing data to Bank 7
		Write_Bank_8_Inq	[24]	Capability of writing data to Bank 8
		Write_Bank_9_Inq	[25]	Capability of writing data to Bank 9
		Write_Bank_10_Inq	[26]	Capability of writing data to Bank 10
		Write_Bank_11_Inq	[27]	Capability of writing data to Bank 11
		Write_Bank_12_Inq	[28]	Capability of writing data to Bank 12
		Write_Bank_13_Inq	[29]	Capability of writing data to Bank 13
		Write_Bank_14_Inq	[30]	Capability of writing data to Bank 14
		Write_Bank_15_Inq	[31]	Capability of writing data to Bank 15
80008h	LUT_Ctrl	Presence_Inq	[0]	Presence of this Feature 0: Not Available, 1: Available
			[1-4]	Reserved
		ON_OFF	[5]	Read: read a status Write: ON or OFF this feature 0: OFF 1: ON When ON is written, the ON_OFF field of the GAMMA register is turned to OFF.
			[6-27]	Reserved
		Active_Bank	[28-31]	Active bank
8000Ch	Bank_0_Offset_Inq	Bank_0_Quadlet_Offset	[0-31]	32-bit offset of Bank 0 table data
80010h	Bank_1_Offset_Inq	Bank_1_Quadlet_Offset	[0-31]	32-bit offset of Bank 1 table data
80014h	Bank_2_Offset_Inq	Bank_2_Quadlet_Offset	[0-31]	32-bit offset of Bank 2 table data
80018h	Bank_3_Offset_Inq	Bank_3_Quadlet_Offset	[0-31]	32-bit offset of Bank 3 table data
8001Ch	Bank_4_Offset_Inq	Bank_4_Quadlet_Offset	[0-31]	32-bit offset of Bank 4 table data
80020h	Bank_5_Offset_Inq	Bank_5_Quadlet_Offset	[0-31]	32-bit offset of Bank 5 table data

Offset	Name	Field	Bit	Description
80024h	Bank_6_ Offset_Inq	Bank_6_Quadlet_Offset	[0-31]	32-bit offset of Bank 6 table data
80028h	Bank_7_ Offset_Inq	Bank_7_Quadlet_Offset	[0-31]	32-bit offset of Bank 7 table data
8002Ch	Bank_8_ Offset_Inq	Bank_8_Quadlet_Offset	[0-31]	32-bit offset of Bank 8 table data
80030h	Bank_9_ Offset_Inq	Bank_9_Quadlet_Offset	[0-31]	32-bit offset of Bank 9 table data
80034h	Bank_10_ Offset_Inq	Bank_10_Quadlet_Offset	[0-31]	32-bit offset of Bank 10 table data
80038h	Bank_11_ Offset_Inq	Bank_11_Quadlet_Offset	[0-31]	32-bit offset of Bank 11 table data
8003Ch	Bank_12_ Offset_Inq	Bank_12_Quadlet_Offset	[0-31]	32-bit offset of Bank 12 table data
80040h	Bank_13_ Offset_Inq	Bank_13_Quadlet_Offset	[0-31]	32-bit offset of Bank 13 table data
80044h	Bank_14_ Offset_Inq	Bank_14_Quadlet_Offset	[0-31]	32-bit offset of Bank 14 table data
80048h	Bank_15_ Offset_Inq	Bank_15_Quadlet_Offset	[0-31]	32-bit offset of Bank 15 table data

7.6.3 Example: Setting Gamma Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.7 Saturation

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

Saturation is applicable to color models only.

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

7.7.1 SATURATION: 814h

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 147)*.

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

Field	Bit	Description
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.

7.7.2 Example: Setting Saturation Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.8 Hue

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

Hue is applicable to color models only.

7.8.1 HUE: 810h

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 147).

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.

7.8.2 Example: Setting Hue Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.9 Sharpness

The camera 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 Off by default. Otherwise, the default setting is On.

7.9.1 SHARPNESS: 808h

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 147)*.

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.

7.9.2 Example: Setting Sharpness Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.10 White Balance

The camera 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 Auto and One Push white balance. One use of One Push/Auto 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 One Push/Auto will result 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.

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

7.10.1 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.

7.10.2 Example: Setting White Balance Using the FlyCapture API

The following FlyCapture 2.0 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 );
```

7.11 Bayer Color Processing

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

To determine the actual pattern on your camera, query the BAYER_TILE_MAPPING register 1040h (page 114).

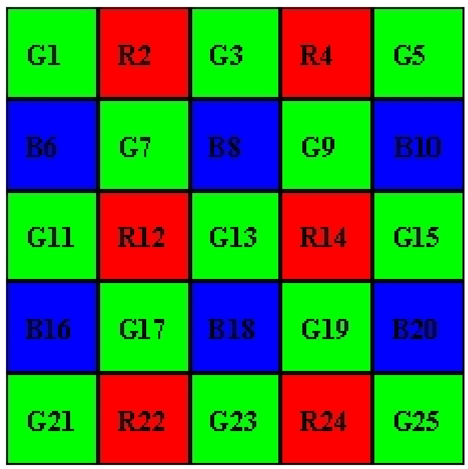


Figure 7.1: Example Bayer Tile Pattern

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:

$$Y = R/4 + G/2 + B/4$$

7.11.1 Accessing Raw Bayer Data

Users interested in accessing the raw Bayer data to apply their own color conversion algorithm or one of the FlyCapture library algorithms should acquire images using one of the Format 7 video modes that support Raw pixel encoding. (See Supported Formats, Modes and Frame Rates on page 67.)

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

7.11.2 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.

Because color models support on-board color processing, the camera reports YYYY tiling when operating in any non-raw Bayer data format. For more information, see Bayer Color Processing on previous page.

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.

7.11.3 Example: Accessing Raw Bayer Data using FlyCapture2

Using the FlyCapture 2 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)

7.12 Image Flip/Mirror

The camera supports horizontal image mirroring. The mirror image operation is performed on the camera using the on-board frame buffer (page 30).

7.12.1 MIRROR_IMAGE_CTRL: 1054h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature. 0: Not Available, 1: Available
	[1-30]	Reserved.
Mirror_Image_Ctrl	[31]	Value 0: Disable horizontal (mirror) image flip 1: Enable horizontal (mirror) image flip

7.13 High Dynamic Range (HDR) Imaging

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

The FlyCapture SDK includes the HighDynamicRange example program. This illustrates how to use the API to capture images in HDR mode.

7.13.1 HDR: 1800h – 1884h

This register allows the user to access and control a multiple exposure quick cycle mode, which is useful for high dynamic range (HDR) imaging.

Note that if bit [31] of the FRAME_INFO register 12F8h (page 118) is set to 1, the camera will embed the current shutter/gain value in the image when bit [6] of HDR_CTRL is set to 1. The image timestamp will be embedded in the first 32-bits of image data, the shutter value in the second 32-bits, and gain in the third, all in big-endian format.

Note that the on/off bit for the HDR shutter and gain registers is hard-coded to on.

Format:

Offset	Name	Field	Bit	Description
1800h	HDR_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
1820h	HDR_SHUTTER_0	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
		-	[1-19]	Reserved
		Value	[20-31]	Query SHUTTER_INQ register 51Ch for range of possible shutter values
1824h	HDR_GAIN_0	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
		-	[1-19]	Reserved
		Value	[20-31]	Query GAIN_INQ register 520h for range of possible gain values
1840h	HDR_SHUTTER_1	Same format as HDR_SHUTTER_0		
1844h	HDR_GAIN_1	Same format as HDR_GAIN_0		
1860h	HDR_SHUTTER_2	Same format as HDR_SHUTTER_0		
1864h	HDR_GAIN_2	Same format as HDR_GAIN_0		
1880h	HDR_SHUTTER_3	Same format as HDR_SHUTTER_0		
1884h	HDR_GAIN_3	Same format as HDR_GAIN_0		

7.14 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 operating in Y8 (8bits/pixel) mode, this is therefore 4 pixels worth of data.

The following frame-specific information can be provided:

- Timestamp
- Gain
- Shutter
- Brightness
- Exposure
- 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`):

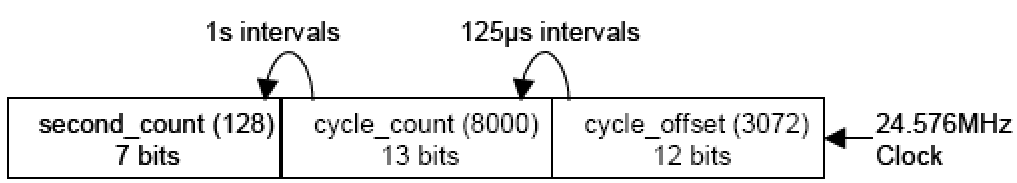
- data[0] = first byte of Timestamp data
- data[4] = first byte of Gain data
- data[24] = first byte of Frame Counter data

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 black and white cameras, white balance is still included, but no valid data is provided.

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 3071, which 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.

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.

7.14.1 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 118)
	[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 117)

8 Troubleshooting

8.1 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.

8.2 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.

8.2.1 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

8.2.2 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.

8.2.3 TIME_FROM_BUS_RESET: 12E4h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Time_From_Reset	[1-31]	Time in seconds since the camera detected a bus reset .

8.2.4 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.

8.2.5 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.

8.2.6 CAMERA_LOG: 1D00 – 1DFFh

Format:

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

8.3 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.

Table 8.1: LED During Camera Power-up and Operation

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

8.3.1 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

8.4 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.

Enabling raw Bayer output when operating in a monochrome data format *(page 64)* *produces an image shift effect in the test pattern.*

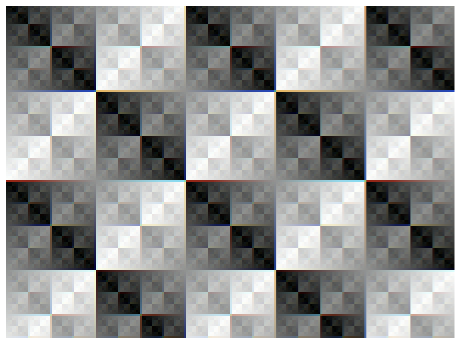


Figure 8.1: Test Pattern Sample Image

8.4.1 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

8.5 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).

8.5.1 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 (page 64).

Related Knowledge Base Articles

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

8.5.2 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

8.6 Channel Balancing

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

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

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

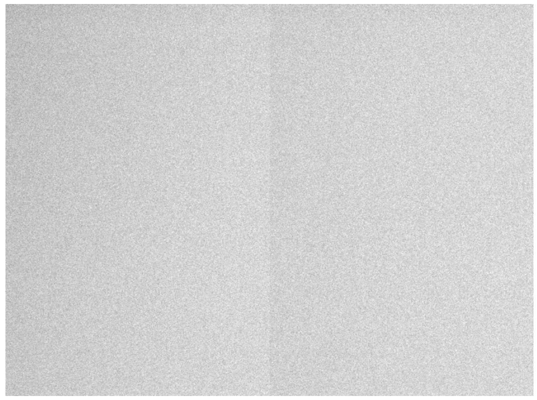


Figure 8.2: Example of dual channel image with no balancing

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

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

Balancing is only done in full resolution modes.

8.7 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.

8.7.1 Smear Reduction

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

- Reducing 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 100ns 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

Appendix A: Control and Status Registers

A.1 General Register Information

A.1.1 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.

A.1.2 Config ROM

A.1.2.1 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

A.1.2.2 Unit Directory

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

A.1.2.3 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

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

A.1.3 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 page 147). 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;
```

A.2 Inquiry Registers

A.2.1 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

A.2.2 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
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

Offset	Name	Field	Bit	Description
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

A.2.3 Video Frame Rate Inquiry Registers

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

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

Offset	Name	Field	Bit	Description
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)	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
20Ch	V_RATE_INQ_0_3 (Format 0, 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]	240 FPS
			[8-31]	Reserved
210h	V_RATE_INQ_0_4 (Format 0, 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]	120 FPS
		FrameRate_7	[7]	240 FPS
			[8-31]	Reserved

Offset	Name	Field	Bit	Description
214h	V_RATE_INQ_0_5 (Format 0, Mode 5)	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
218h	V_RATE_INQ_0_6 (Format 0, Mode 6)	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
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)	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]	Reserved
			[8-31]	Reserved
228h	V_RATE_INQ_1_2 (Format 1, Mode 2)	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
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

Offset	Name	Field	Bit	Description
234h	V_RATE_INQ_1_5 (Format 1, Mode 5)	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
238h	V_RATE_INQ_1_6 (Format 1, Mode 6)	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
23Ch	V_RATE_INQ_1_7 (Format 1, Mode 7)	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
240h	V_RATE_INQ_2_0 (Format 2, Mode 0)	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

Offset	Name	Field	Bit	Description
244h	V_RATE_INQ_2_1 (Format 2, 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]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
248h	V_RATE_INQ_2_2 (Format 2, Mode 2)	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
24Ch	V_RATE_INQ_2_3 (Format 2, 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]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
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

Offset	Name	Field	Bit	Description
254h	V_RATE_INQ_2_5 (Format 2, Mode 5)	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
258h	V_RATE_INQ_2_6 (Format 2, Mode 6)	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
25Ch	V_RATE_INQ_2_7 (Format 2, Mode 7)	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
260h : 2BFh	Reserved			
2E0h	V_CSR_INQ_7_0	Mode_0	[0-31]	32-bit offset for Format 7 Mode 0
2E4h	V_CSR_INQ_7_1	Mode_1	[0-31]	32-bit offset for Format 7 Mode 1
2E8h	V_CSR_INQ_7_2	Mode_2	[0-31]	32-bit offset for Format 7 Mode 2
2ECh	V_CSR_INQ_7_3	Mode_3	[0-31]	32-bit offset for Format 7 Mode 3
2F0h	V_CSR_INQ_7_4	Mode_4	[0-31]	32-bit offset for Format 7 Mode 4
2F4h	V_CSR_INQ_7_5	Mode_5	[0-31]	32-bit offset for Format 7 Mode 5
2F8h	V_CSR_INQ_7_6	Mode_6	[0-31]	32-bit offset for Format 7 Mode 6
2FCh	V_CSR_INQ_7_7	Mode_7	[0-31]	32-bit offset for Format 7 Mode 7

Offset	Name	Field	Bit	Description
300h	V_CSR_INQ_7_8	Mode_8	[0-31]	32-bit offset for Format 7 Mode 8
304h	V_CSR_INQ_7_9	Mode_9	[0-31]	32-bit offset for Format 7 Mode 9
308h	V_CSR_INQ_7_10	Mode_10	[0-31]	32-bit offset for Format 7 Mode 10
30Ch	V_CSR_INQ_7_11	Mode_11	[0-31]	32-bit offset for Format 7 Mode 11
310h	V_CSR_INQ_7_12	Mode_12	[0-31]	32-bit offset for Format 7 Mode 12
314h	V_CSR_INQ_7_13	Mode_13	[0-31]	32-bit offset for Format 7 Mode 13
318h	V_CSR_INQ_7_14	Mode_14	[0-31]	32-bit offset for Format 7 Mode 14
31Ch	V_CSR_INQ_7_15	Mode_15	[0-31]	32-bit offset for Format 7 Mode 15
320h	V_CSR_INQ_7_16	Mode_16	[0-31]	32-bit offset for Format 7 Mode 16
324h	V_CSR_INQ_7_17	Mode_17	[0-31]	32-bit offset for Format 7 Mode 17
328h	V_CSR_INQ_7_18	Mode_18	[0-31]	32-bit offset for Format 7 Mode 18
32Ch	V_CSR_INQ_7_19	Mode_19	[0-31]	32-bit offset for Format 7 Mode 19
330h	V_CSR_INQ_7_20	Mode_20	[0-31]	32-bit offset for Format 7 Mode 20
334h	V_CSR_INQ_7_21	Mode_21	[0-31]	32-bit offset for Format 7 Mode 21
338h	V_CSR_INQ_7_22	Mode_22	[0-31]	32-bit offset for Format 7 Mode 22
33Ch	V_CSR_INQ_7_23	Mode_23	[0-31]	32-bit offset for Format 7 Mode 23
340h	V_CSR_INQ_7_24	Mode_24	[0-31]	32-bit offset for Format 7 Mode 24
344h	V_CSR_INQ_7_25	Mode_25	[0-31]	32-bit offset for Format 7 Mode 25
348h	V_CSR_INQ_7_26	Mode_26	[0-31]	32-bit offset for Format 7 Mode 26
34Ch	V_CSR_INQ_7_27	Mode_27	[0-31]	32-bit offset for Format 7 Mode 27
350h	V_CSR_INQ_7_28	Mode_28	[0-31]	32-bit offset for Format 7 Mode 28
354h	V_CSR_INQ_7_29	Mode_29	[0-31]	32-bit offset for Format 7 Mode 29
358h	V_CSR_INQ_7_30	Mode_30	[0-31]	32-bit offset for Format 7 Mode 30
35Ch	V_CSR_INQ_7_31	Mode_31	[0-31]	32-bit offset for Format 7 Mode 31

A.2.4 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.

A.2.5 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 (see the Advanced Registers section) from the base address of initial register space. (Vendor unique)

Offset	Name	Field	Bit	Description
484h	PIO_Control_CSR_Inq	PIO_Control_ Quadlet_Offset	[0-31]	32-bit offset of the PIO control CSRs (see the Parallel Input/Output (PIO) section) 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 (see the Serial Port Input/Output (SIO) section) from the base address of initial register space.
48Ch	Strobe_Output_CSR_Inq	Strobe_Output_ Quadlet_Offset	[0-31]	32-bit offset of the strobe output signal CSRs (see the Strobe Signal Output section) 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.

A.2.6 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		

Offset	Name	Field	Bit	Description
52Ch	TEMPERATURE_INQ	Same format as the BRIGHTNESS_INQ register		
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		

Offset	Name	Field	Bit	Description
540h : 57Ch	Reserved for other FEATURE_HI_INQ			
580h	ZOOM_INQ	Same format as the BRIGHTNESS_INQ register		
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		

A.3 Video Mode Control and Status Registers

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

A.3.1 FORMAT_7_RESIZE_INQ: 1AC8h

This register reports all internal camera processes being used to generate images in the current Format 7 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
H_Bin	[29]	Standard horizontal binning (addition of adjacent lines within horizontal shift register) 0: Off, 1: On

Field	Bit	Description
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

A.3.2 Inquiry Registers for Custom Video Mode (Format 7) Offset Addresses

The following set of registers indicates the locations of the custom video mode (Format 7) base registers. These offsets are relative to the base offset 0xFFFF F0F0 0000.

Table A.1: Custom Video Mode (Format 7) Inquiry Register Offset Addresses

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

A.3.2.1 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)

A.3.2.2 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

A.3.2.3 PACKET_PARA_INQ, BYTE_PER_PACKET, and PACKET_PER_FRAME

If the *Presence* bit in the VALUE_SETTING register (page 147) is zero, values of these fields will be updated by writing the new value to the IMAGE_POSITION, IMAGE_SIZE (page 144) and COLOR_CODING_ID (page 145) registers with the value of the ISO_Speed register (page 75).

First, the ISO_Speed register must be written. Then the IMAGE_POSITION, IMAGE_ SIZE and COLOR_CODING_ID registers should be updated.

If the *Presence* bit in the VALUE_SETTING register is one, the values of these fields will be updated by writing one to the *Setting_1* bit in the VALUE_SETTING register. If the *ErrorFlag_1* bit is zero after the *Setting_ 1* bit returns to zero, the values of these fields are valid.

Format:

Address	Name	Field	Bit	Description
Base + 034h	PIXEL_NUMBER_ INQ	PixelPerFrame	[0-31]	Total number of pixels in the required image area
Base + 038h	TOTAL_BYTES_HI_ INQ	BytesPerFrameHi	[0-31]	Higher 32-bits of total bytes of image data per frame
Base + 03Ch	TOTAL_BYTES_LO_ INQ	BytesPerFrameLo	[0-31]	Lower 32-bits of total bytes of image data per frame
Base + 040h	PACKET_PARA_INQ	UnitBytePerPacket	[0-15]	Minimum bytes per packet; packet sizes must be multiples of the minimum
		MaxBytePerPacket	[16-31]	Maximum bytes per packet
Base + 044h	BYTE_PER_PACKET	BytePerPacket	[0-15]	Packet size. This value determines the real packet size and transmission speed for one frame image. BytePerPacket = UnitBytePerPacket * n BytePerPacket <= MaxBytePerPacket If BytePerPacket * n != Bytes Per Frame‡, you must use padding. (n is an integer)
		RecBytePerPacket	[16-31]	Recommended bytes per packet. If this value is zero, ignore this field.
Base + 048h	PACKET_PER_ FRAME_INQ	PacketPerFrame	[0-31]	Number of packets per frame. Updated after BytePerPacket is written. Total number of bytes of transmission data per one frame = BytePerPacket * PacketPerFrame Number of bytes of padding = BytePerPacket * PacketPerFrame - Bytes Per Frame‡. The receiver must ignore the above padding in the last packet of each frame.

‡Example: Bytes Per Frame = Resolution Size * 1 byte per pixel = 640 * 480 = 307200 bytes per frame

A.3.2.4 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 147) If 0, the camera can't report the value and it should be ignored.

A.3.2.5 VALUE_SETTING

The purpose of the *Setting_1* bit is for updating the TOTAL_BYTES_HI_INQ, TOTAL_BYTES_LO_INQ, PACKET_PARA_INQ and BYTE_PER_PACKET (page 146) registers. If one of the values in the IMAGE_POSITION, IMAGE_SIZE (page 144) COLOR_CODING_ID (page 145) and ISO_SPEED (page 75) registers is changed, the *Setting_1* bit must be set to 1.

Format:

Address	Name	Field	Bit	Description
Base + 07Ch	VALUE_SETTING	Presence	[0]	If this bit is 1, Setting_1 , ErrorFlag_1 and ErrorFlag_2 fields are valid. This bit is read only.
		Setting_1	[1]	If writing "1" to this bit, IMAGE_POSITION, IMAGE_SIZE, COLOR_CODING_ID and ISO_Speed register value will be reflected in PIXEL_NUMBER_INQ, TOTAL_BYTES_HI_INQ, TOTAL_BYTES_LO_INQ, PACKET_PARA_INQ and BYTE_PER_PACKET registers. This bit is self-cleared.
			[2-7]	Reserved
		ErrorFlag_1	[8]	Combination of the values of IMAGE_POSITION, IMAGE_SIZE, COLOR_CODING_ID and ISO_Speed register is not acceptable. 0: no error, 1: error This flag will be updated every time Setting_1 bit returns to "0" from "1".
		ErrorFlag_2	[9]	BytePerPacket value is not acceptable. 0: no error, 1: error Updated after BytePerPacket value is written. If 0, transmission can be started.
			[10-31]	Reserved

A.4 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.

A.4.1 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.

A.4.2 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
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.

A.4.3 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
Hue	Hue	deg	degree	0	Relative
Saturation	Saturation	%		100	Relative
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

A.4.4 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.*

Appendix B: 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.

B.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 B.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>

B.2 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
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		

Mode	Video Format	240fps	120fps	60fps	30fps	15fps	7.5fps	3.75fps	1.875fps
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

Mode	Video Format	120fps	60fps	30fps	15fps	7.5fps	3.75fps	1.875fps
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

B.3 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 B.2: Isochronous Data Packet Format for Format 7

0-7	8-15	16-23		24-31	
data_length		tag	channel	tCode	sy
header_CRC					
Video data payload					
data_CRC					

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.

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

Revision History

Revision	Date	Notes
1.0	August 29, 2011	Initial version–support for model GX-FW-10K3
2.0	December 19, 2011	Added support for model GX-FW-28S5 Minor edits and clarifications
2.1	January 3, 2012	Minor change to introductory language.
3.0	April 9, 2012	Added support for model GX-FW-60S6 Added installation instructions Added information on smear reduction
3.1	October 4, 2012	Clarification of offset registers Updated Imaging Performance specifications and QE graphs Clarified sensor package glass for GX-FW-10K3

Index