Grasshopper2 GS2-FW

Technical Reference Manual

Version 1.0

Revised 8/29/2011

Point Grey Research® Inc.

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

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

You are cautioned that any changes or modifications not expressly approved in this manual could void your authority to operate this equipment.

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 two (2) 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

1 Introduction

The fully redesigned Grasshopper2 camera is the next generation version of the high performance Grasshopper. With a selection of the same image sensors and housed in the same form factor as the Grasshopper, the Grasshopper2 offers several new features, including enhanced opto-isolated GPIO and improved imaging performance.

All model-specific information presented in this manual reflects functionality available in firmware version 1.6.3.0.

To check the camera firmware version, consult our knowledge base: www.ptgrey.com/support/kb/index.asp?a=4&q=94.

Related Knowledge Base Articles

ID	Title	URL
206	Key features and benefits of the IEEE-1394b standard	www.ptgrey.com/support/kb/index.asp?a=4&q=206

1.1 Using This Manual

This manual provides the user with a detailed specification of the camera system. The reader should be aware that the camera system is complex and dynamic – if any errors or omissions are found during experimentation, please contact us. (See Appendix G: Contacting Point Grey Research on page 163.)

This document is subject to change without notice.

1.2 GS2-FW-14S5 Specifications

1.2.1 General Features and Specifications

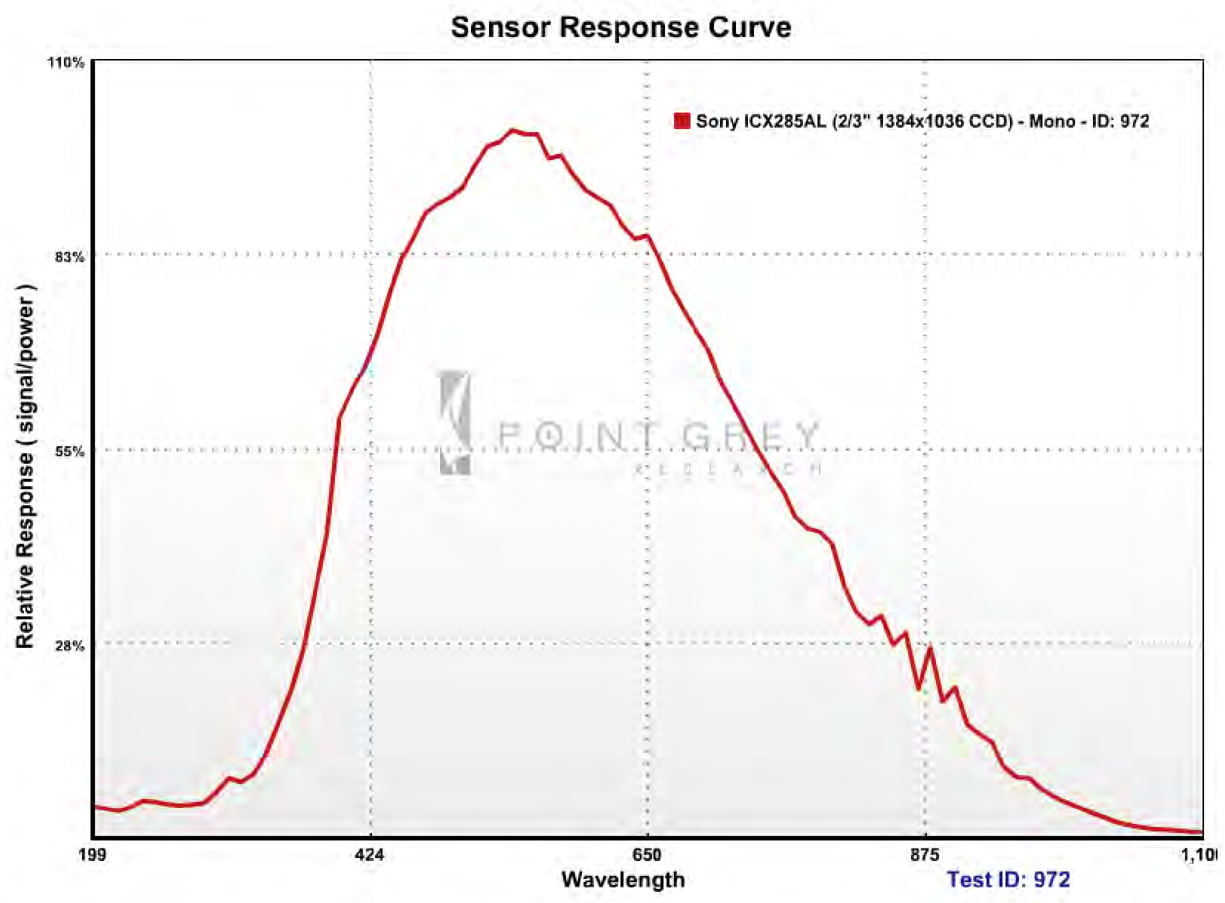
Imaging Sensor	Sony® ICX 285 2/3"progressive scan CCD
Max Pixels	1384 x 1036
Pixel Size	6.45 µm x 6.45 µm
Pixel Clock Frequency	28 MHz maximum
A/D Converter	14-bit
Max FPS at Max Resolution	30 FPS
Video Data Output	8, 12, 16 and 24-bit digital data
Partial Image Modes	Pixel binning and region of interest modes available via Format_7
Gain	Automatic/Manual/One-Push Gain modes
	-3.6 dB to 24 dB
Shutter	Automatic/Manual/One-Push/Extended Shutter modes
	0.03 ms to 330 seconds (extended shutter mode)
Gamma	0.50 to 4.00
Full Well Depth*	19,845 e- at zero gain
Signal To Noise Ratio*	64 dB
Dark Noise*	10.79 e-/s at zero gain
Dark Current*	204 e-/s at zero gain
Read Noise*	12.06 e- at zero gain
Trigger Modes	IIDC Trigger Modes 0, 1, 3, 4, 5, 14 and 15
Lens Mount	C-mount
Interfaces	9-pin IEEE-1394b for camera control and video data transmission 8-pin HR25 female GPIO for power, trigger and strobe
Camera Specification	IIDC 1394-based Digital Camera Specification v1.32
Power Requirements	Voltage: 8-30 V Power: less than 2.5 W
Dimensions	44 mm x 29 mm x 58 mm (without optics)
Mass	104 grams (without optics)
Emissions Compliance	Complies with CE rules and Part 15 Class A of FCC Rules

*Based on Format_7 Mode_0, Mono16 pixel format, max resolution, 0 dB gain, 1.0 ms exposure.

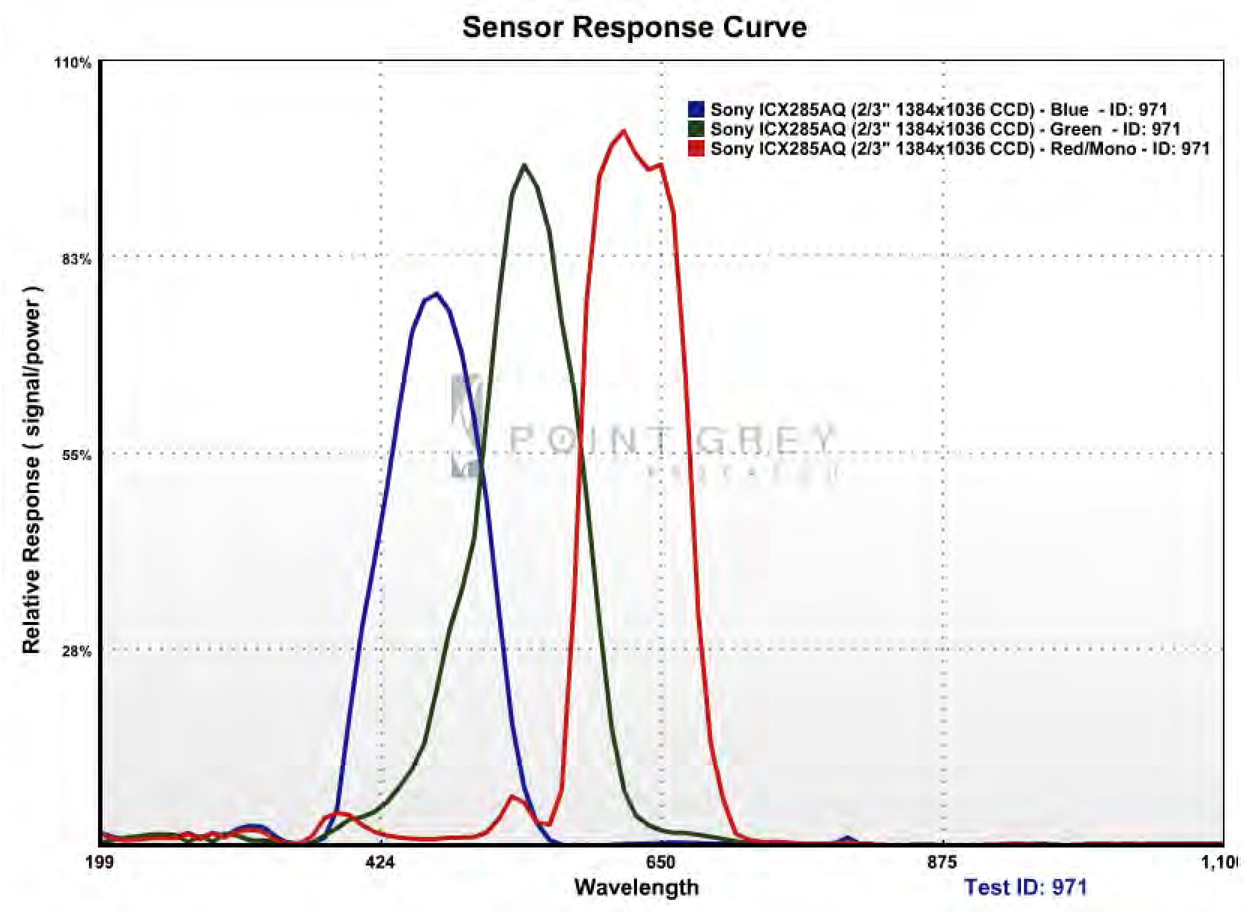
Operating Temperature	Commercial grade electronics rated from 0° to 45°C
Operating Relative Humidity	20 to 80% (no condensation)
Storage Temperature	-30° to 60°C
Storage Relative Humidity	20 to 95% (no condensation)
Warranty	Two years

1.2.2 GS2-FW-14S5 Sensor Response

GS2-FW-14S5 (Mono)



GS2-FW-14S5 (Color)



1.3 Analog-to-Digital Conversion

The camera sensor incorporates an Analog Devices 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 right-most bits are always zero. The following table illustrates the most important aspects of the processor. For more information, refer to the Analog Devices website at www.analog.com.

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

2 General Camera Operation

2.1 Handling Precautions and Camera Care

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

Your Point Grey digital camera module 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. Please consult the following knowledge base article for more details: www.ptgrey.com/support/kb/index.asp?a=4&q=42.
- Users who have purchased a bare board camera should take the following additional protective measures:
    - Either handle bare handed or use non-chargeable gloves, clothes or material. Also, use conductive shoes.
    - Install a conductive mat on the floor or working table to prevent the generation of static electricity.
- When handling the camera unit, avoid touching the lenses. Fingerprints will affect the quality of the image produced by the device.
- To clean the lenses, use a standard camera lens cleaning kit or a clean dry cotton cloth. Do not apply excessive force.
- To clean the imaging surface of your sensor, follow the steps outlined in www.ptgrey.com/support/kb/index.asp?a=4&q=66.
- Our cameras are designed for an office environment or laboratory use. 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.

2.1.1 Case Temperature and Heat Dissipation

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. The case can reach temperatures up to 45° Celsius under normal operating conditions. This is expected behavior and will not damage the camera electronics.

If reducing heat is a concern, users can 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.

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 1).

Temp. Sensor Specifications

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

2.2 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 ideal input voltage is 12V DC; however, the camera is designed to handle voltages between 8V and 30V DC. The power consumption is outlined in the Camera Specifications section.

Power can also be provided through the GPIO interface. For more information, see General Purpose Input/Output (GPIO) on page 30. The camera selects whichever power source is supplying a higher voltage.

Some systems - such as laptop computers or those with several FireWire devices connected - require an external power supply to power the camera. For suggestions on how to provide power in these circumstances, consult the following knowledge base article:

KB Article 93: www.ptgrey.com/support/kb/index.asp?a=4&q=93

Components of the camera can be powered-up or powered-down using the CAMERA_POWER register 0x610. The exact components, such as image sensor, A/D converter and other board electronics, vary between camera models. By default, power is OFF both at startup and reinitialization.

If isochronous transmit (ISO_EN (page 45)/ ONE_SHOT / MULTI_SHOT (page 45)) is enabled while the camera is powered down, the camera will automatically write *Cam_Pwr_Ctrl* = 1 to 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 Configuration Sets on page 18.

CAMERA_POWER: 610h

Format:

Field	Bit	Description
Cam_Pwr_Ctrl	[0]	Write: 0: Begin power-down process 1: Begin power-up process 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
	[1-30]	Reserved
Camera_Power_Status	[31]	Read only Read: the pending value of Cam_Pwr_Ctrl

2.3 Managing Camera Settings

2.3.1 Using the 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 1394-based Digital Camera Specification v1.32. *Format* tables for each 32-bit register are presented along with their respective feature descriptions throughout this manual. These tables describe the purpose of each bit that comprises the register. Bit 0 is always the most significant bit of the register value.

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

For more information about camera registers, including the base register memory map, config ROM offsets and calculating register addresses, see General Register Information on page 123

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

Mode:
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.

One_Push	ON_OFF	A_M_Mode	State
N/A	0	N/A	**Off state.**

POINT GREY

One_Push	ON_OFF	A_M_Mode	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	1	0	**One-Push action**. Camera controls feature by itself only once and returns to the Manual control state with adjusted value.

Table 2.1: CSR Mode Control Descriptions

Value:
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.

2.3.1.1 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 on page 127:

- *Inquiry Registers for Basic Functions and Feature Presence*: To determine if a particular function or feature is available on the camera.
- *Inquiry Registers for Feature Elements*: To determine if elements of a particular feature are available on the camera.
- *Video Format, Mode and Frame Rate Inquiry Registers*: 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 Format_7 Video Modes* (page 144)
- *Inquiry Registers for Serial I/O* (page 75)
- *Inquiry Registers for Lookup Table Functionality* (page 97)

2.3.1.2 Using the Absolute Value Registers

Many Point Grey IEEE-1394 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 intererest (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 151.

2.3.2 Operating System and Software Support

2.3.2.1 Macintosh and Linux OS Support

Users wishing to operate their Point Grey camera on the Macintosh OS/X or Linux operating systems should consult the following knowledge base articles:

Macintosh support:

www.ptgrey.com/support/kb/index.asp?a=4&q=173

Linux support:

www.ptgrey.com/support/kb/index.asp?a=4&q=330
www.ptgrey.com/support/kb/index.asp?a=4&q=17

2.3.2.2 FlyCap Demo Program

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

2.3.2.3 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.

Third-Party Software Applications

The following knowledge base article provides information on Point Grey IEEE-1394 camera compatibility with third-party software development kits, applications, camera drivers, and integrated development environments (IDEs):

KB Article 152: www.ptgrey.com/support/kb/index.asp?a=4&q=152

2.3.3 User Configuration Sets

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 is used for the default factory settings that users can always restore to. 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.

For a listing of all registers saved, see Memory Channel Registers on next page.

MEMORY_SAVE: 618h

Format:

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

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.

CUR_MEM_CH: 624h

Format:

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

2.3.3.1 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

Register Name	Offset
WHITE_BALANCE	80Ch
HUE	810h
SATURATION	814h
GAMMA	818h
SHUTTER	81Ch
GAIN	820h
IRIS	824h
FOCUS	828h
TRIGGER_MODE	830h
TRIGGER_DELAY	834h
FRAME_RATE	83Ch
PAN	884h
TILT	888h
ABS_VAL_AUTO_EXPOSURE	908h
ABS_VAL_SHUTTER	918h
ABS_VAL_GAIN	928h
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

Register Name	Offset
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

2.4 Camera Error and Status Monitoring

2.4.1 Status Indicator LED

LED Status	Description
Maximum red (Initial connection)	Initial startup. On until camera is initialized.
Maximum red (During operation)	Condition 1: Bus Rest. On for 0.66s. Condition 2: Power failure. On until power-up via CAMERA_POWER 0x610 (page 16).
Dull Red	Configuration error.
Bright Red	Configuration error.
Dull Green	Camera is idle.
Bright Green	Firewire activity. On for 0.5s during activity.
Dull Yellow	Powered down.
Bright Yellow	Powered down + activity. On for 0.5s during activity.
Red/Green flashing	Camera firmware is being updated. Flashes at 5Hz.

Table 2.2: Status indicator LED descriptions

LED_CTRL: 1A14h

This register allows the user to turn off the camera's status LED. LED(s) are re-enabled the next time the camera is power cycled.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-22]	Reserved
LED_Ctrl	[23-31]	Enable or disable the LED 0x00: Off, 0x74: On

2.4.2 General Status Monitoring

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

TIME_FROM_INITIALIZE: 12E0h

This register reports the time, in seconds, since the camera (FPGA) was initialized. This initialization occurs during a hard power-up. This is different from powering up the camera via the CAMERA_POWER (page 16) register, which will not reset this time.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
Time_From_Init	[1-31]	Time in seconds since the camera was initialized.

TIME_FROM_BUS_RESET: 12E4h

This register reports the time, in seconds, since the last bus reset occurred. This will be equal to the value reported by TIME_FROM_INITIALIZE (page 22) if no reset has occurred since the last time the camera was initialized.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
Time_From_Reset	[1-31]	Time in seconds since the camera detected a bus reset .

VOLTAGE: 1A50h – 1A54h

This register allows the user to access and monitor the various voltage registers supported by the camera.

Format:

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

CURRENT: 1A58h-1A5Ch

This register allows the user to access and monitor the various current registers supported by the camera.

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]	Quadlet offset of the current CSR's, which report the current in amps using the 32-bit floating-point IEEE/REAL*4 format.

TEMPERATURE: 82Ch

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. *Value* is in kelvins (0°C = 273.15K) in increments of one-tenth (0.1) of a kelvin.

For more information about camera temperature, see Case Temperature and Heat Dissipation on page 14.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A, 1: Available
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.
	[2-4]	Reserved

Field	Bit	Description
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Write: 1: Begin to work (self-cleared after operation) Read: 0: Not in operation, 1: In operation If A_M_Mode = 1, this bit is ignored
ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Read: read a current mode 0: Manual, 1: Automatic
	[8-19]	Reserved
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

2.4.3 Error Status Registers

XMIT_FAILURE: 12FCh

This register 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.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
Frame_Count	[1-31]	Read: Count of failed frame transmissions. Write: Reset.

VMODE_ERROR_STATUS: 628h

This register is used by the camera to report any camera configuration errors. If an error has occurred, no image data will be sent by the camera.

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.

CAMERA_LOG: 1D00 – 1DFFh

This register provides access to the camera's 256 byte internal message log, which is often useful for debugging camera problems. Characters are hexadecimal representations of ASCII characters. Contact technical support for interpretation of message logs.

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.

2.4.4 Device Information

PIXEL_CLOCK_FREQ: 1AF0h

This register 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.

Format:

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

HORIZONTAL_LINE_FREQ: 1AF4h

This register specifies the current horizontal line frequency in Hz in IEEE-754 32-bit floating point format.

Format:

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

SERIAL_NUMBER: 1F20h

This register specifies the unique serial number of the camera.

Format:

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

MAIN_BOARD_INFO: 1F24h

This register specifies the type of camera (according to the main printed circuit board).

Format:

Field	Bit	Description	
Major_Board_Design	[0-11]	0x2: Digiclops 0x3: Dragonfly 0x4: Sync Unit 0x6: Ladybug Head 0x7: Ladybug Base Unit 0x8: Bumblebee 0xA: Scorpion Back Board 0x10: Flea 0x12: Dragonfly Express 0x18: Dragonfly2	0x19: Flea2 0x1A: Firefly MV 0x1C: Bumblebee2 0x1F: Grasshopper 0x22: Grasshopper2 0x21: Flea2G-13S2 0x24: Flea2G-50S5 0x26: Chameleon 0x2B: Flea3 0x36: Zebra2
Minor_Board_Rev	[12-15]	Internal use	
Reserved	[16-31]	Reserved	

SENSOR_BOARD_INFO: 1F28h

This register specifies the type of imaging sensor used by the camera (due to the wide variety of sensors available).

The interpretation of this register varies depending on the camera type, as defined in the MAIN_BOARD_INFO register 0x1F24 (page 25) 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

BUILD_TIMESTAMP: 1F40h

This register specifies the date that the current version of the firmware was built in Unix time format.

Format:

Field	Bit	Description
Build_Date	[0-31]	Date firmware was built (read-only)

FIRMWARE_VERSION: 1F60h

This register contains the version information for the currently loaded camera firmware. For more information on Point Grey versioning standards, see Knowledge Base Article 96 (http://www.ptgrey.com/support/kb/index.asp?a=4&q=96).

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

FIRMWARE_BUILD_DATE: 1F64h

Specifies the date that the current version of the firmware was built in Unix time format.

Format:

Field	Bit	Description
Build_Date	[0-31]	Date firmware was built (read-only)

FIRMWARE_DESCRIPTION: 1F68-1F7Ch

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

2.5 Non-Volatile Flash Memory

The camera has 512 KB of non-volatile memory for users to store data.

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.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-5]	Reserved
Clean_Page	[6]	Write: 1: Write page to data flash 0: No-op Read: 1: Page is clean 0: Page is dirty
	[7]	Reserved
Page_Size	[8-19]	8 == 256 byte page 9 == 512 byte page
Num_Pages	[20-31]	11 == 2048 pages 13 == 8192 pages

DATA_FLASH_DATA: 1244h

This register provides the quadlet offset to the start of the actual data contained in the flash memory (query DATA_FLASH_CTRL register 1240h (page 27)).

Any access outside of a modified page will automatically cause the page to be rewritten to flash, i.e. the user can write as much information as necessary, then perform a single write to the DATA_FLASH_CTRL register 1240h.

Format:

Field	Bit	Description
DF_Data	[0-31]	Quadlet offset to the start of data

2.6 Upgrading Camera Firmware

Camera firmware can be upgraded or downgraded to later or earlier versions using the UpdatorGUI2 program that is bundled with every firmware version available from www.ptgrey.com/support/downloads. 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. For more information on updating camera firmware, consult the UpdatorGUI2 User Manual, included with the firmware bundle.

2.7 Camera Accessories

Accessories such as tripod mounts are available from Point Grey. Contact our Sales team at sales@ptgrey.com for additional information. Links to digital camera accessories can be found in the following knowledge base article:

KB Article 131: www.ptgrey.com/support/kb/index.asp?a=4&q=131.

3 Camera Interface and Connectors

3.1 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 15

For a full description of the features and benefits of 1394b, refer to 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.

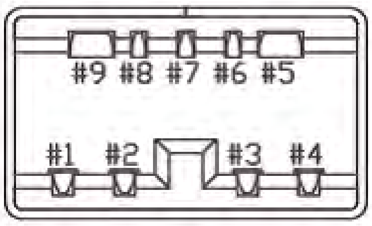


Figure 3.1: IEEE-1394b connector pin configuration (as shown looking at the rear of the camera)

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)

Table 3.1: IEEE-1394b connector pin configuration

3.1.1 Daisy Chaining

As the camera has two (2) IEEE-1394b connectors, it is possible to connect multiple cameras (and/or hubs) in a daisy-chained manner. This configuration makes it easy to connect multiple cameras to a single host controller. However, the maximum bandwidth available for all cameras is still restricted to 800Mbps.

3.2 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. To purchase IEEE-1394-compliant cables from Point Grey, visit http://www.ptgrey.com/products/firepro/index.asp.

Consult the following knowledge base article for information on how to extend the physical distance between the camera and the controlling host system:

KB Article 197: www.ptgrey.com/support/kb/index.asp?a=4&q=197

3.3 Host Adapter Card

Camera kits include a 2-port IEEE-1394 PCI host adapter card or PCI Express Card. For more information regarding the differences between various 1394 host adapters, consult the following knowledge base article:

KB Article 146: www.ptgrey.com/support/kb/index.asp?a=4&q=146

3.4 General Purpose Input/Output (GPIO)

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

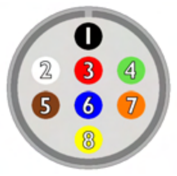
Diagram	Pin	Function	Function
	1	IO0	Opto-isolated input (default Trigger in)
	2	IO1	Opto-isolated output
	3	IO2	Input / Output / serial transmit (TX)
	4	IO3	Input / Output / serial receive (RX)
	5	GND	Ground for bi-directional IO, V_{EXT}, +3.3 V pins
	6	GND	Ground for opto-isolated IO pins
	7	V_{EXT}	Allows the camera to be powered externally
	8	+3.3 V	Power external circuitry up to 150 mA

Table 3.2: GPIO pin assignments (as shown looking at rear of camera)

Inputs can be configured to accept external trigger signals. Outputs can be configured to send an output signal. For information about I/O configuration, refer to the following sections:

- *Asynchronous External Trigger Modes*
- *Programmable Strobe Output*
- *Serial Communication Using GPIO*
- *Pulse Width Modulation (PWM)*

3.4.1 GPIO Electrical Characteristics

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

KB Article 200: www.ptgrey.com/support/kb/index.asp?a=4&q=200

The $\mathbf{V_{EXT}}$ pin (Pin 7) allows the camera to be powered externally. The voltage limit is 8-30 V, and current is limited to 1A.

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

3.4.2 GPIO0 (Opto-Isolated Input) Circuit

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

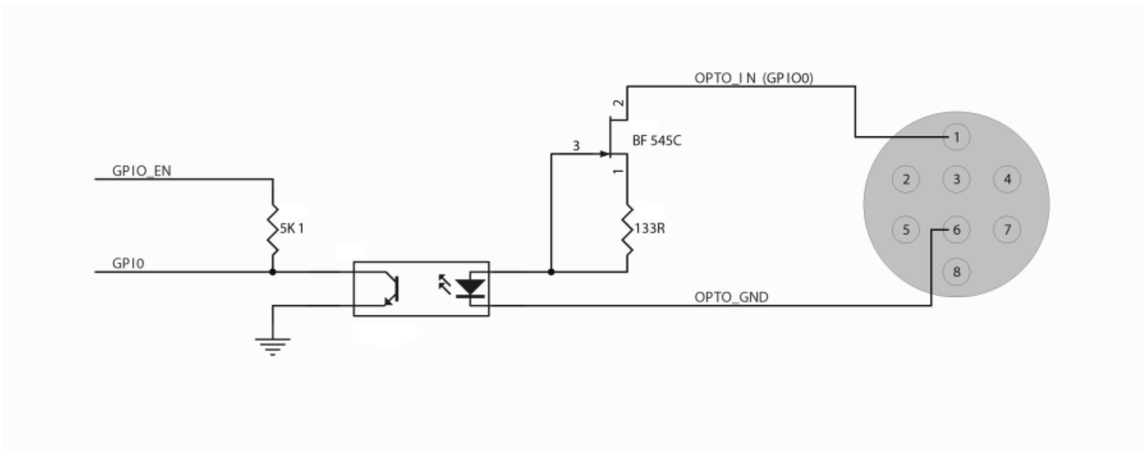


Figure 3.2: Optical input circuit

- *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*

3.4.3 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.

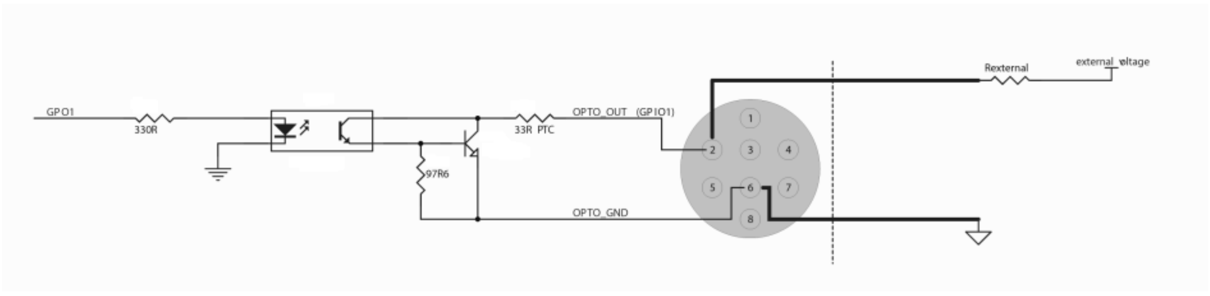


Figure 3.3: Optical output circuit

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
30 V	4.7 kΩ	1.51 V	13.3 mA

3.4.4 GPIO 2 / 3 (Bi-Directional) Circuit

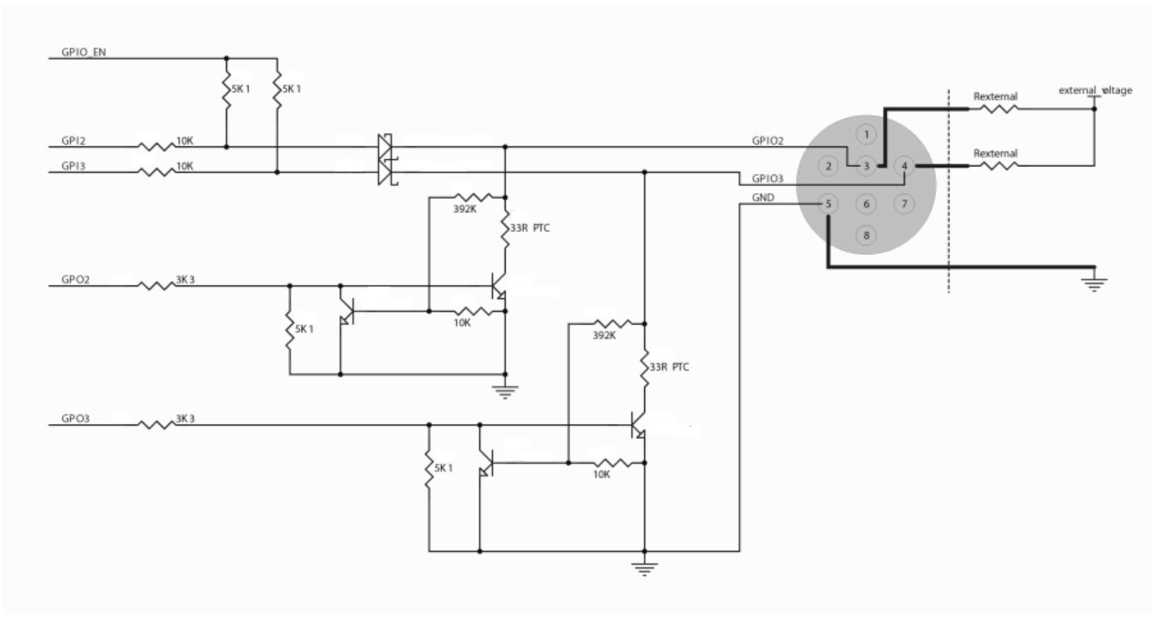


Figure 3.4: Figure 4: GPIO2 / 3 Circuit

3.4.4.1 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.

3.4.4.2 Output Side

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

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
30 V	1 kΩ	0.966 V

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

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

4 Video Formats, Modes and Frame Rates

4.1 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.

4.1.1 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 frame rate, 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 on page 156.

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.

4.1.2 Maximum Number of Cameras on a Single Bus

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. See

Knowledge Base Article 146 for more information.

- 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.

4.2 Custom Video Modes Using Format_7

The camera implements a number of IIDC Format_7 customizable video modes. These modes, which may allow for faster frame rate and/or increased 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 “pixel binning” or "subsampling." Some modes implement a combination of ROI, binning and/or subsampling. On Point Grey cameras, "binning" refers to aggregation that takes place in an analog manner, directly on the sensor before read-out. "Subsampling" refers to aggregation that takes place digitally on the FPGA, after read-out. Unless specified otherwise, color data is maintained in pixel binning/subsampling modes.

The figures below illustrate how binning or subsampling work in general. 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.

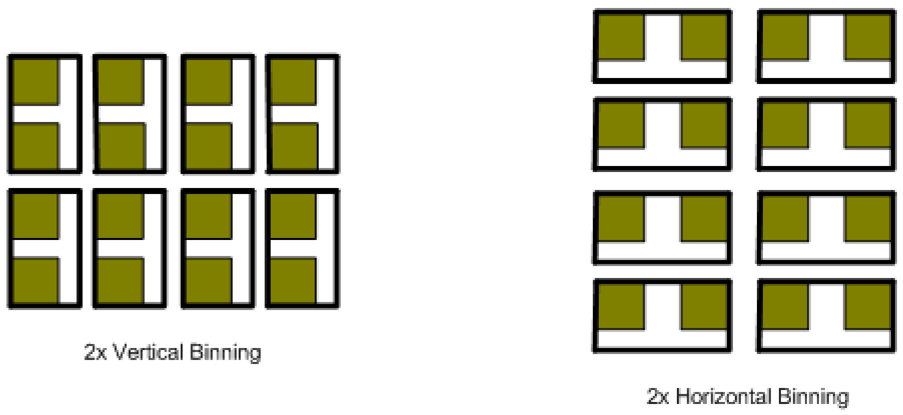


Figure 4.1: 2x Vertical and 2x Horizontal Binning

Some binning operations average the pixel values after aggregation. When this happens, there is usually little or no change in overall image intensity. This type of binning is referred to as "binning plus averaging." Other binning operations do not perform any averaging after aggregation, which usually results in increased intensity. Binning without averaging is referred to as "additive binning."

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. Changing the size of the image or the pixel encoding format requires the stop/start procedure. 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.

Additional binning information can be obtained by reading the FORMAT_7_RESIZE_INQ (page 145) 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.

For information about configuring the camera in Format_7 mode, see Video Format and Mode CSRs on next page

For information about configuring Format_7 modes/sizes and Format_7-related inquiry registers, see Video Mode Control and Status Registers for Format_7 on page 144.

When operating in Format_7 mode, the Feature_Lo_Inq register 408h reports the presence of the Pan and Tilt features. However, these feature are off and cannot be turned on.

4.2.1 Exceeding Bandwidth Limitations Using Format_7

There is a mechanism for effectively bypassing IEEE-1394 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 IIDC modes. For additional information, see Knowledge Base Article 256.

4.2.2 Format_7 Mode Descriptions

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. However, overall image quality may be poorer than in Mode_7. 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 on the FPGA. In monochrome models, vertical binning is performed on the sensor, and horizontal binning is performed on the FPGA. 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 color models, however, no frame rate increase is achieved.

Mode_3

Mode_3 is an optimized imaging performance mode that operates by disabling the CCD pixel gain amplifier and slowing the vertical pixel clock shift register. These features result in reduced dark current and very long shutter times.

When operating in this mode, keep in mind the following:

- This mode implements 2X horizontal and 2X vertical additive binning, and is available only when outputting to a monochrome pixel format.
- Due to the slower pixel clock, image exposure and readout do not overlap. Although frame rates are compromised, the longest extended shutter times (up to 330 s) are available.
- The noise reduction feature of this mode is not available when operating in an asynchronous trigger mode.

Mode_4

Mode_4 implements a combination of 2X horizontal and 2X vertical binning plus averaging, and is available on color models only. 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_8

Mode_8 is similar to Mode_3, except this mode does not perform pixel binning, and is available in all pixel encoding formats.

4.3 Video Format and Mode CSRs

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

FRAME_RATE: 83Ch

This register 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 CURRENT_ FRAME_RATE register 600h (page 39). 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 register is set to OFF when the camera is operating in asynchronous trigger mode. For more information, see TRIGGER_MODE: 830h on page 59.

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 the Absolute Value CSRs section of the Appendix (page 151).

Format:

Same format as BRIGHTNESS: 800h on page 87.

Related Resources

Title	Link
FlyCapture SDK *ExtendedShutterEx* sample program	ExtendedShutterEx

CURRENT_FRAME_RATE: 600h

Allows the user to query and modify the current frame rate of the camera.

Format:

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

CURRENT_VIDEO_MODE: 604h

Allows the user to query and modify the current video mode of the camera.

Format:

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

CURRENT_VIDEO_FORMAT: 608h

Allows the user to query and modify the current video format of the camera.

Format:

Field	Bit	Description
Cur_V_Format	[0-2]	Current video format Format_0 .. Format_7 ***Note***: GigE Vision cameras operate only in Format_7 mode.
	[3-31]	Reserved.

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

The following FlyCapture 2.0 code snippet sets the camera to one of the IIDC standard formats: 640x480 Y8 at 60 FPS.

```
Camera.SetVideoModeandFrameRate( VIDEOMODE_640x480Y8 , FRAMERATE_
60 );
```

4.4 GS2-FW-14S5 Standard Formats, Modes and Frame Rates

4.4.1 Standard Formats, Modes and Frame Rates

Models: 🔵 **14S5C** ● **14S5M**

Modes	1.875fps	3.75fps	7.5fps	15fps	30fps
640x480 YUV411	🔵	🔵	🔵	🔵	🔵
640x480 YUV422	🔵	🔵	🔵	🔵	🔵
640x480 RGB	🔵	🔵	🔵	🔵	🔵
640x480 Y8	🔵●	🔵●	🔵●	🔵●	🔵●
640x480 Y16	🔵●	🔵●	🔵●	🔵●	🔵●
1280x960 YUV 422	🔵	🔵	🔵	🔵	
1280x960 RGB	🔵	🔵	🔵	🔵	
1280x960 Y8	🔵●	🔵●	🔵●	🔵●	
1280x960 Y16	🔵●	🔵●	🔵●	🔵●	

Table 4.1: Supported video formats, modes and frame rates (GS2-FW-14S5)

4.4.2 Format_7 Modes and Frame Rates

Mode	Pixel Format	Max Size (HxV)	Unit Size (H,V)	Max Res	1280x 960	640x 480	320x 240	160x 120
0	Mono8	1384x1036	8,2	30	31	48	66	82
0	Mono12	1384x1036	8,2	30	31	48	66	82
0	Mono16	1384x1036	8,2	26	30	48	66	82
0	Raw8	1384x1036	8,2	29	31	48	66	82
0	Raw12	1384x1036	8,2	29	31	48	66	82
0	Raw16	1384x1036	8,2	26	30	48	66	82
0	RGB8	1384x1036	8,2	18	20	48	66	82
0	YUV411	1384x1036	8,2	30	31	48	66	82
0	YUV422	1384x1036	8,2	26	30	48	66	82
0	YUV444	1384x1036	8,2	18	20	48	66	82
1	Mono8	692x518	4,2	29	-	31	48	66
1	Mono12	692x518	4,2	29	-	31	48	66
1	Mono16	692x518	4,2	29	-	31	48	66
1	Raw8	692x518	4,2	29	-	31	48	66
1	Raw12	692x518	4,2	29	-	31	48	66
1	Raw16	692x518	4,2	29	-	31	48	66
1	RGB8	692x518	4,2	29	-	31	48	66
1	YUV411	692x518	4,2	29	-	31	48	66
1	YUV422	692x518	4,2	29	-	31	48	66
1	YUV444	692x518	4,2	29	-	31	48	66
3	Mono8	692x518	-	11	-	-	-	-
3	Mono12	692x518	-	11	-	-	-	-
3	Mono16	692x518	-	11	-	-	-	-
4	Mono8	692x518	4,2	48	-	50	66	76
4	Mono12	692x518	4,2	48	-	50	66	76
4	Mono16	692x518	4,2	48	-	50	66	76
8	Mono8	1384x1036	-	6	-	-	-	-
8	Mono12	1384x1036	-	6	-	-	-	-
8	Mono16	1384x1036	-	6	-	-	-	-
8	Raw8	1384x1036	-	6	-	-	-	-
8	Raw12	1384x1036	-	6	-	-	-	-
8	Raw16	1384x1036	-	6	-	-	-	-

Mode	Pixel Format	Max Size (HxV)	Unit Size (H,V)	Max Res	1280x 960	640x 480	320x 240	160x 120
8	RGB8	1384x1036	-	6	-	-	-	-
8	YUV411	1384x1036	-	6	-	-	-	-
8	YUV422	1384x1036	-	6	-	-	-	-
8	YUV444	1384x1036	-	6	-	-	-	-

Table 4.2: Supported partial image (Format 7) video formats and modes for GS2-FW-14S5

5 Image Acquisition and Transmission

5.1 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 1394 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 156)

5.1.1 Camera Power and Isochronous Transmission

If isochronous transmit (ISO_EN (page 45)/ ONE_SHOT / MULTI_SHOT (page 45)) is enabled while the camera is powered down, the camera will automatically write *Cam_Pwr_Ctrl* = 1 to 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 Configuration Sets on page 18.

5.1.2 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 88 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 FlyCapture 2.0 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, read the FRAME_INFO register 12F8h (page 109) for the parameters applied to a specific frame. Note, however, that this register reads property values that have been written by the end of shutter integration, even if they are not yet in effect.

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 = 100Mbps 1 = 200Mbps 2 = 400Mbps
	[8-15]	Reserved
Operation_Mode	[16]	1394 operation mode Change control register sets of ISO_Channel and ISO_Speed

Field	Bit	Description
		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 = 100Mbps 1 = 200Mbps 2 = 400Mbps 3 = 800Mbps 4 = 1.6Gbps 5 = 3.2Gbpss

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

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.

5.2 High Dynamic Range (HDR) Imaging

The camera can be set into a High Dynamic Range mode in which it rotates 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 format of the HDR registers is as follows:

Offset	Register	Remarks
0x1800	HDR control register	Toggle bit [6] to enable/disable HDR
0x1820	HDR shutter register for image 0	Similar to SHUTTER register 0x81C
0x1824	HDR gain register for image 0	Similar to GAIN register 0x820
0x1840	HDR shutter register for image 1	Similar to SHUTTER register 0x81C
0x1844	HDR gain register for image 1	Similar to GAIN register 0x820
0x1860	HDR shutter register for image 2	Similar to SHUTTER register 0x81C
0x1864	HDR gain register for image 2	Similar to GAIN register 0x820
0x1880	HDR shutter register for image 3	Similar to SHUTTER register 0x81C
0x1884	HDR gain register for image 3	Similar to GAIN register 0x820

Note that the on/off bit (bit [6]) for the HDR shutter and gain registers is hard-coded to on.

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 109) 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 quadlet of image data, the shutter value in the second quadlet, and gain in the third, all in big-endian format.

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]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields will be read only
		-	[7-31]	Reserved

Offset	Name	Field	Bit	Description
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		

5.3 Pixel Formats

5.3.1 Y16 (16-bit Mono) Image Acquisition

The camera can output Y16 (16 bit-per-pixel) mono images. Because the camera uses a 14-bit A/D converter (page 13) , only 14 bits of data are useable. Unused bits are always zero.

To determine the number of bits of useable image data, and resulting signal-to-noise ratio, that is actually being produced by the A/D converter, see www.ptgrey.com/support/kb/index.asp?a=4&q=170.

The data format for Y16 images is controlled by the DATA_DEPTH register 630h (page 47).

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.

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. Write: Ignored Read: Effective data depth
Little_Endian	[8]	Little endian mode for 16-bit pixel formats only Has no effect if not in a 16-bit pixel format Write/Read: 0: Big endian mode (default on initialization) 1: Little endian mode
	[9-31	Reserved

5.3.2 Y8 or Y16 Raw Bayer Output

When operating in Y8 or Y16 mode, color models can output either grayscale or raw Bayer data. This output is controlled by the BAYER_MONO_CTRL register 1050h (page 48).

BAYER_MONO_CTRL: 1050h

Selecting a half-width, half-height image size and monochrome pixel format, such as 800x600 Y8, using non-Format_7 modes provides a monochrome binned image. In some cases, enabling raw Bayer output in mono mode provides a raw Bayer region of interest of 800x600, centered within the larger pixel array. This has an effect on the field of view.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature. 0: N/A, 1: Available
	[1-30]	Reserved.
Bayer_Mono_Ctrl	[31]	Value 0: Disable raw bayer output in non-Format_7 mono modes 1: Enable raw bayer output in non-Format_7 mono modes

5.4 Automatic Inter-Camera Synchronization

Multiple Point Grey FireWire cameras, when they are on the same IEEE-1394 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 125µs (microseconds) of each other (note: 125µs is the maximum deviation). However, the 1394 bandwidth limits the maximum number of cameras that can be on one bus. See the section Maximum Number of Cameras on a Single 1394 Bus for more information.
- 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 125µs. 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 60).

5.5 Asynchronous Triggering

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

Color models operating in Video Mode_4 (page 37) support Trigger_Mode_4 and Trigger_Mode_5 in monochrome pixel formats only.

5.5.1 External Trigger Timing

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

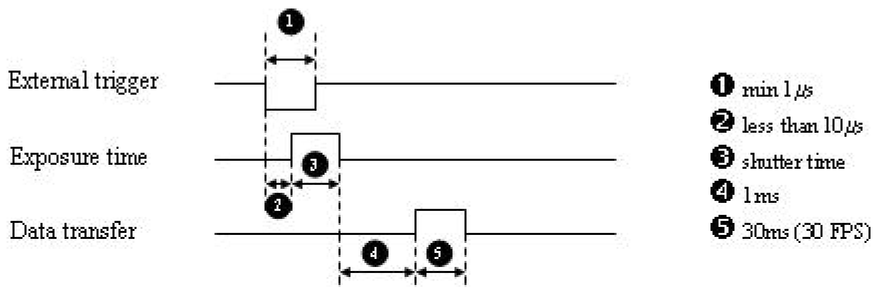


Figure 5.1: External trigger timing characteristics

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 61) 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.

5.5.2 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 25).

5.5.3 Maximum Frame Rate in External Trigger Mode

Historically, the maximum triggered frame rate has been limited by the camera's inability to overlap image transfer with the trigger input. This limitation applies to trigger modes 0, 1, 3, 4 and 5, where supported. The theoretical maximum triggered frame rate in these modes depends on the shutter time (in seconds) and the maximum frame rate of the camera in free-running mode (in Hz).

This relationship is calculated as follows:

Max_Frame_Rate_Trigger = 1 / (Shutter + (1 / Max_Frame_Rate_Free_Running))

For example, consider a camera that can acquire 640x480 images at 30Hz in normal “free-running” mode, at a shutter speed of 0.0020s:

Max_Frame_Rate_Triggered = 1 / (0.0020 + (1 / 30)) = 28.30Hz

In contrast, trigger modes 14 and 15, where supported, overlap trigger input with image readout. These modes support external triggering at close to full frame rate.

5.5.4 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.

5.5.5 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.

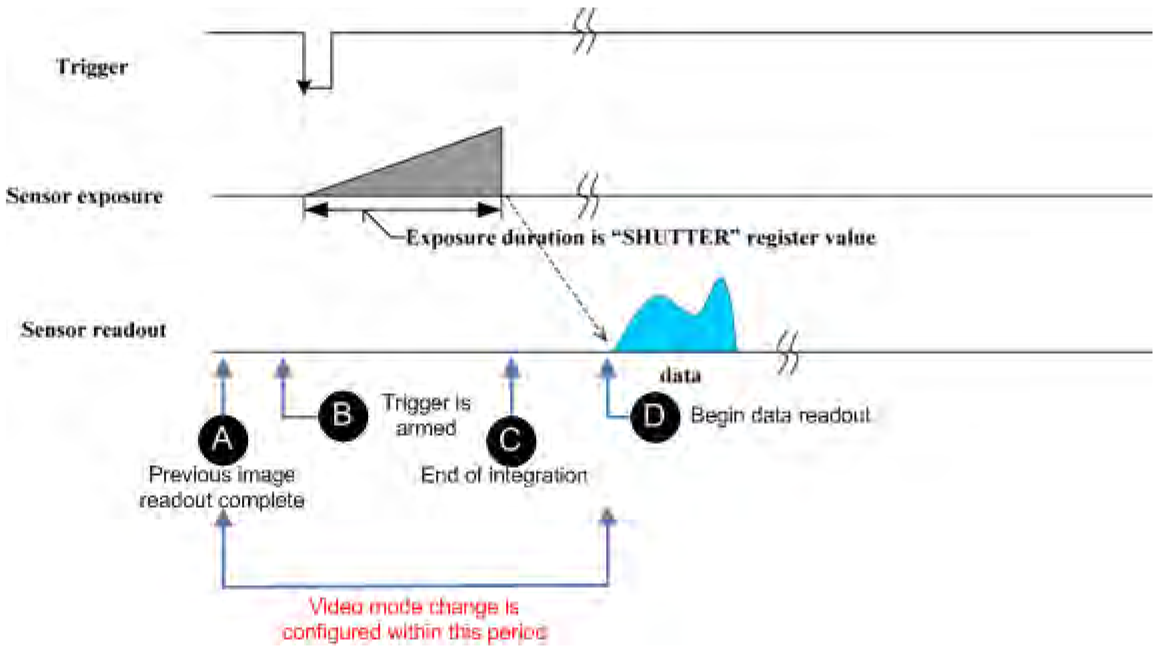

Trigger
Sensor exposure
Exposure duration is "SHUTTER" register value
Sensor readout
data
A
Previous image readout complete
B
Trigger is armed
C
End of integration
D
Begin data readout
Video mode change is configured within this period

Figure 5.2: Relationship Between External Triggering and Video Mode Change Request

When operating in trigger mode 0 (page 51) or trigger mode 1 (page 52), 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 54), 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 55), change requests made after point A for any given image readout are honored only after a delay of one image.

5.5.6 Supported Trigger Modes

5.5.6.1 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 register describes integration time. No parameter is required. The camera can be triggered in this mode using the GPIO pins as external trigger or the SOFTWARE_TRIGGER (62Ch) register.

It is not possible to trigger the camera at full frame rate using Mode_0; however, this is possible using Trigger_Mode_14.

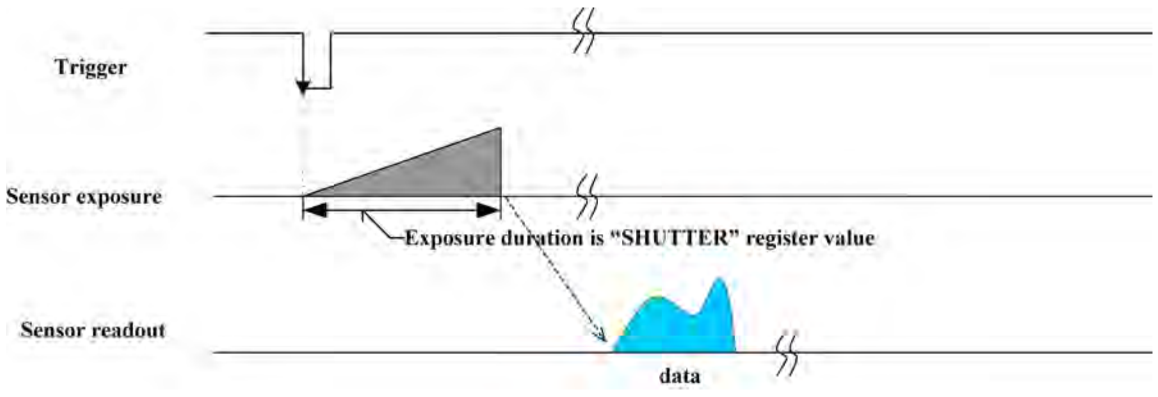


Figure 5.3: Trigger_Mode_0 ("Standard External Trigger Mode")

5.5.6.2 Trigger_Mode_1 ("Bulb Shutter Mode")

Also known as Bulb Shutter mode, the camera starts integration of the incoming light from external trigger input falling edge. Integration time is equal to low state time of the external trigger input.

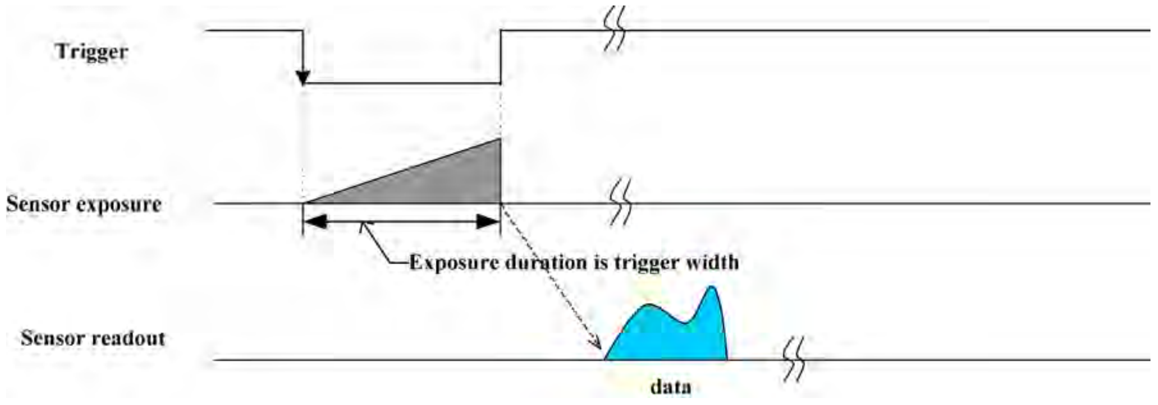


Figure 5.4: Trigger_Mode_1 ("Bulb Shutter Mode")

5.5.6.3 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 bits [20-31] of the TRIGGER_MODE register (offset 830h), the camera will issue a trigger internally at a cycle time that is N times greater than the current frame rate. Again, the SHUTTER register 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.

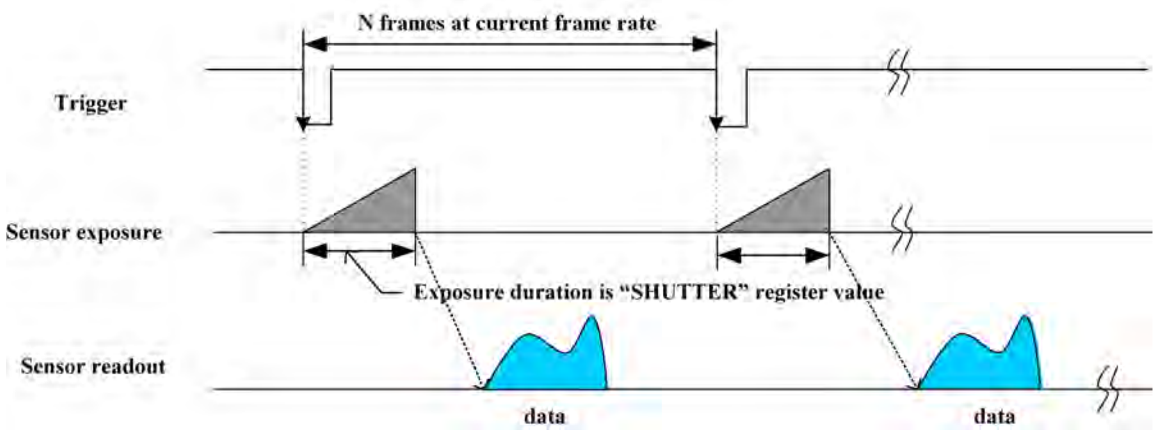

N frames at current frame rate
Trigger
Sensor exposure
Exposure duration is "SHUTTER" register value
Sensor readout
data
data

Figure 5.5: Trigger_Mode_3 ("Skip Frames Mode")

5.5.6.4 Trigger_Mode_4 ("Multiple Exposure Preset Mode")

Trigger_Mode_4 allows the user to set the number of triggered images to be exposed before the image readout starts. In the case of Trigger_Mode_4, the shutter time is controlled by the SHUTTER CSR value; the minimum resolution of the duration is therefore limited by the shutter resolution.

In the figure below, the camera starts integration of incoming light from the first external trigger input falling edge and exposes incoming light at shutter time. Repeat this sequence for N (parameter) external trigger inputs edge then finish integration. Parameter is required and shall be one or more (N >= 1).

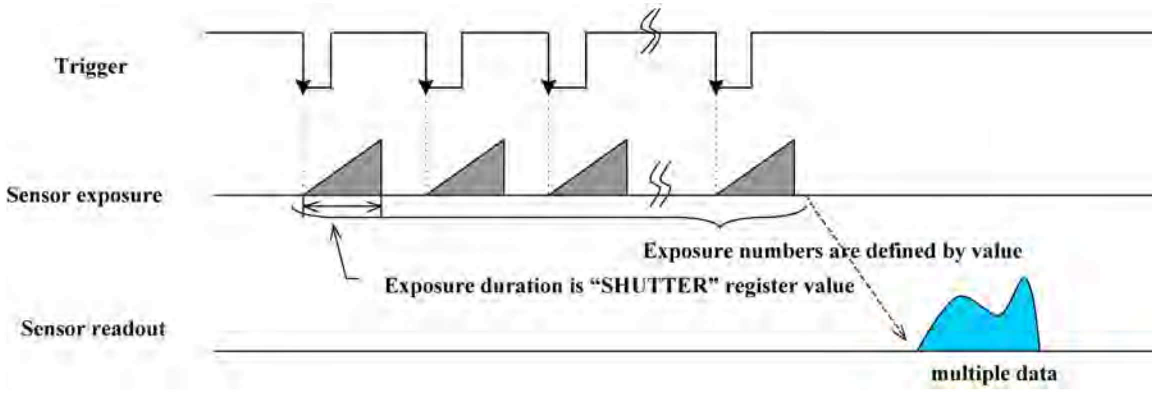

Trigger
Sensor exposure
Exposure numbers are defined by value
Exposure duration is "SHUTTER" register value
Sensor readout
multiple data

Figure 5.6: Trigger_Mode_4 ("Multiple Exposure Preset Mode")

5.5.6.5 Trigger_Mode_5 ("Multiple Exposure Pulse Width Mode")

Trigger_Mode_5 allows the user to set the number of triggered images to be exposed before the image readout starts. In the case of Trigger_Mode_5, the shutter time is controlled by the trigger pulse duration; the minimum resolution of the duration is generally 1 tick of the pixel clock (see PIXEL_CLOCK_FREQ: 1AF0h on page 25). The resolution also depends on the quality of the input trigger signal and the current TRIGGER_DELAY (page 60).

In the figure below, the camera starts integration of incoming light from the first external trigger input falling edge and exposes incoming light until the trigger is inactive. Repeat this sequence for N (parameter) external trigger inputs then finish integration. Parameter is required and shall be one or more (N >= 1).

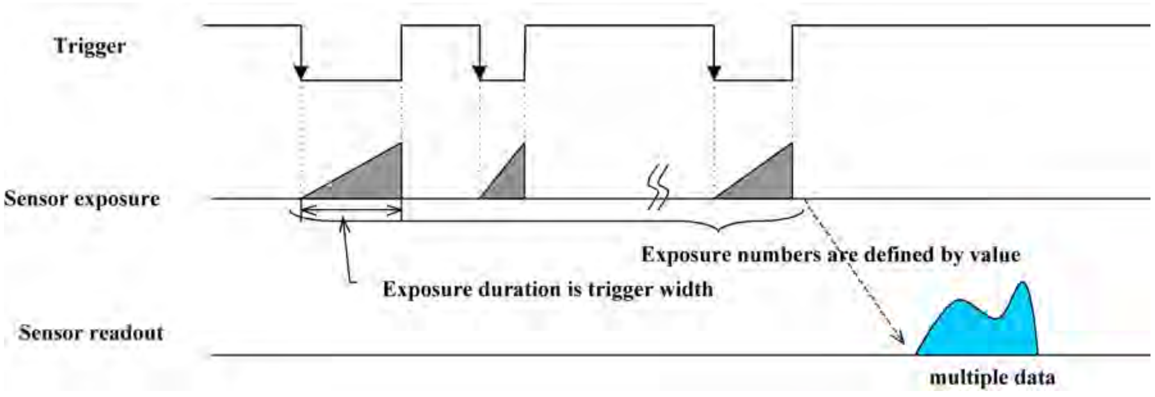

Trigger
Sensor exposure
Exposure numbers are defined by value
Exposure duration is trigger width
Sensor readout
multiple data

Figure 5.7: Trigger_Mode_5 (“Multiple Exposure Pulse Width Mode”)

5.5.6.6 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*(page 57)), 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.

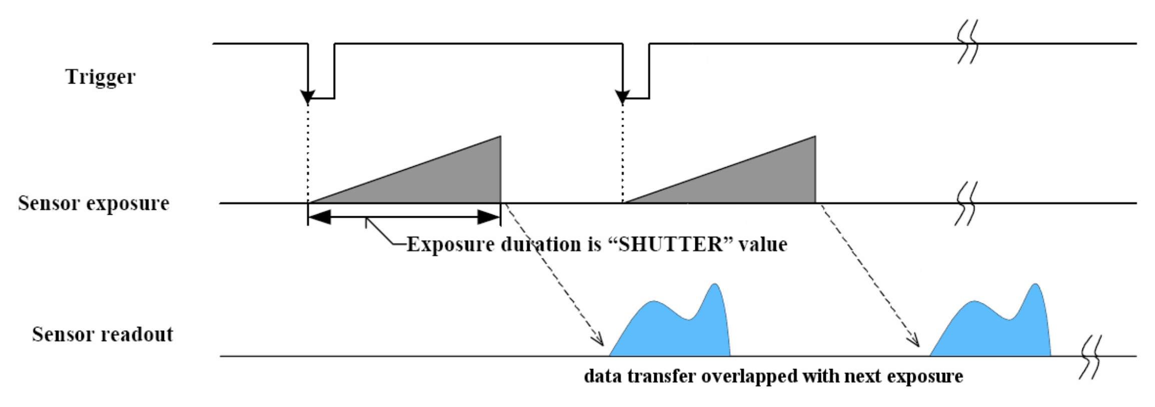

Trigger
Sensor exposure
Exposure duration is “SHUTTER” value
Sensor readout
data transfer overlapped with next exposure

Figure 5.8: Trigger_Mode_14 (“Overlapped Exposure / Readout Mode”)

5.5.6.7 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, 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.

Note: 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 (see Knowledge Base Article 166). 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.

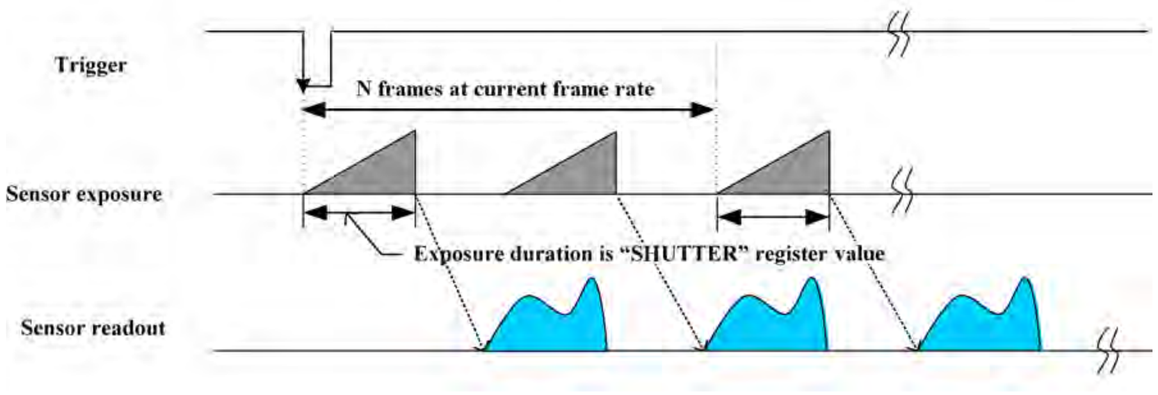

Trigger
N frames at current frame rate
Sensor exposure
Exposure duration is “SHUTTER” register value
Sensor readout

Figure 5.9: Trigger_Mode_15 (“Multi-Shot Trigger Mode”)

5.5.7 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 51).

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. 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 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 60) 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 59). 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 59):

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 61). 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.

5.5.8 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 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);
```

5.5.9 Asynchronous Software Triggering

Shutter integration can be initiated by a register write (software trigger) via SOFTWARE_TRIGGER register 0x62C (page 61).

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

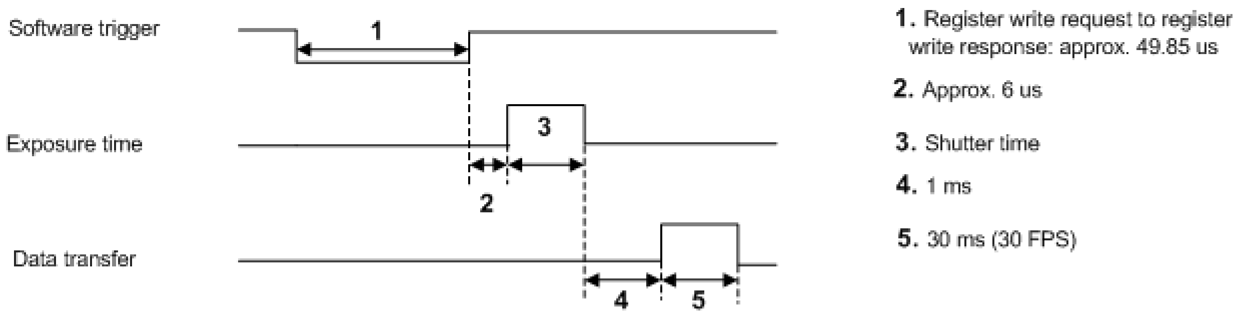


Figure 5.10: Software trigger timing

The time from when the SOFTWARE_TRIGGER register 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 49.85us. The "register 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 6us for a camera capturing 640x480 images. Therefore, the total time from when the register is written to the start of integration is approximately 56us.

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.

5.5.10 Asynchronous Trigger Registers

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: 830h

This register controls the trigger mode. Control of the register is via the *ON_OFF* bit and the *Trigger_ Mode* and *Parameter* fields.

When the ON_OFF bit is set to ON (enabling trigger mode), the FRAME_RATE register (page 38) is taken out of Auto control state and turned to Off state. This change affects the maximum shutter time. For more information, see Extended Shutter Times (page 88). Additionally, if Auto exposure is enabled (page 92), maximum frame rate may be reduced. When the ON_OFF bit is set back to OFF, the FRAME_RATE register is not automatically turned back on or returned to Auto control state.

The *Trigger_Source* bit is used to select which GPIO pin will be used for external trigger purposes.

The *Trigger_Value* bit is used to determine the current raw signal value on the pin.

The *Trigger_ Mode* bit is used to set the trigger mode to be used. For more information, see Supported Trigger Modes on page 51.

Format

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A, 1: Available
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.
	[2-5]	Reserved
ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 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 Sets trigger source ID from *Trigger_ Source_ Inq* field of TRIGGER_INQ register (page 140).
Trigger_Value	[11]	Trigger input raw signal value Read only 0: Low, 1: High
	[8-11]	Reserved

Field	Bit	Description
Trigger_Mode	[12-15]	Trigger mode (Trigger_Mode_0..15) Sets the trigger mode. 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)

TRIGGER_DELAY: 834h

This register provides control over the time delay between one of the following, depending on the current mode:

1. Asynchronous trigger mode: controls the delay between the trigger event and the start of integration (shutter open).
2. Continuous shot (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. For more information, see Automatic Inter-Camera Synchronization on page 48.

Delay is in units of a 24.576 MHz clock. Less than 1024 ticks is linear; greater than 1024 ticks is non-linear. Consider using register 950h ABS_VAL_TRIGGER_DELAY(page 152)

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A, 1: Available
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.
	[2-5]	Reserved
ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields will be read only
	[7-19]	Reserved
Value	[20-31]	Value.

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

Field	Bit	Description
		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

SOFTWARE_TRIGGER: 62Ch

This register allows the user to generate a software asynchronous trigger.

Format:

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

5.6 External Device Control

5.6.1 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)

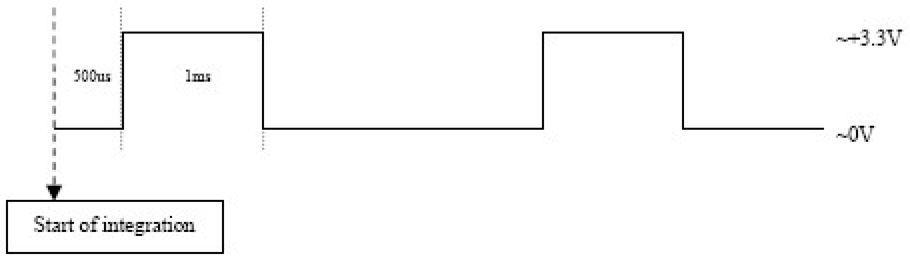
Related Knowledge Base Articles

ID	Title	URL
200	Buffering a GPIO pin strobe output signal using an optocoupler to drive external devices	www.ptgrey.com/support/kb/index.asp?a=4&q=200
212	GPIO strobe signal continues after isochronous image transfer stops	www.ptgrey.com/support/kb/index.asp?a=4&q=212

5.6.1.1 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 60). 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 68); 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:

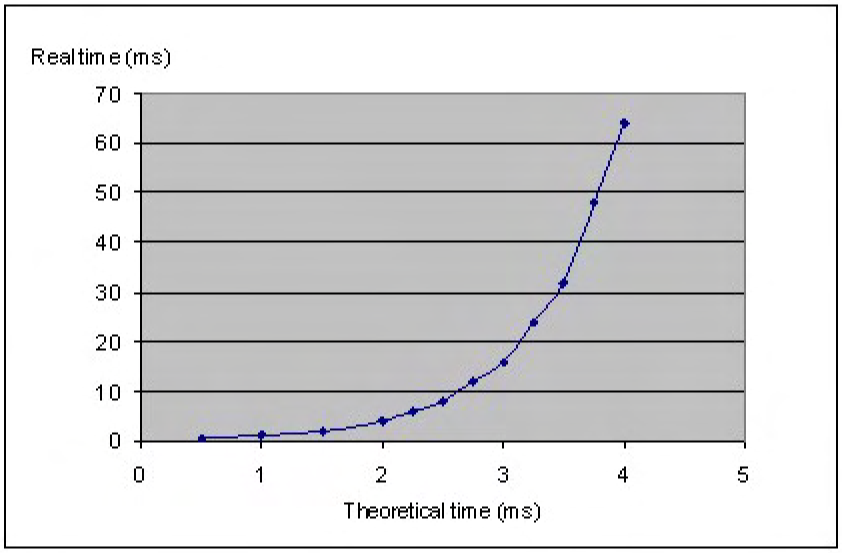
8	E	0	0	0	F	F	F	Hex
1000	1110	0000	0000	0000	1111	1111	1111	Binary
0-7		8-15		16-23		24-31		Bits

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 68).

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

Duration_Value / Delay_Value	Real Time (ms)
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
```

5.6.1.2 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);
```

5.6.1.3 Example: Setting GPIO Pins to Output a Strobe Pattern

Consider the following example strobe pattern scenario for two cameras on the same IEEE-1394 bus:

Parameter	Value
Camera model	
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.

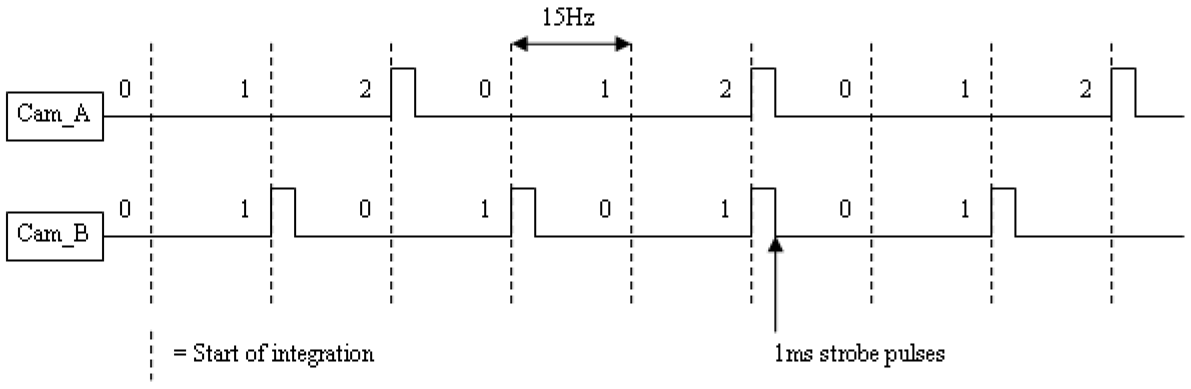


Figure 5.11: Example of multiple camera strobe pattern synchronization

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.

Configure the Desired Pins to Output a Strobe

See *Example: Setting a GPIO Pin To Strobe* on page 1 for instructions on configuring each camera to output a 1ms pulse off a GPIO pin.

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 70). 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.

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 71).

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(page 65)If they are not aligned correctly, simply adjust the *Enable_Mask* until the pulses are synchronized.

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 70) into the image by setting Bit [24] of the FRAME_INFO register 0x12F8 (page 109). 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

}
```

5.6.1.4 Strobe Signal Output Registers

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

Inquiry Register for Strobe Output CSR Offset Addresses

The following register indicates the locations of the Strobe Output CSR registers. These offsets are relative to the base offset 0xFFFF F0F0 0000.

Offset	Name	Field	Bit	Description
48Ch	STROBE_OUTPUT_CSR_INQ	Strobe_ Output_ Quadlet_Offset	[0-31]	Quadlet offset of the Strobe output signal CSRs from the base address of initial register space

Current Strobe Output Register Offsets

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

Format:

Offset	Name	Field	Bit	Description
1300h	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
1304h : 13FCh	Reserved			
1400h	STROBE_0_ INQ	Presence_Inq	[0]	Presence of this feature
0: N/A 1: Available			[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

Offset	Name	Field	Bit	Description
			[7]	Reserved
		Min_Value	[8-19]	Minimum value for this feature control
		Max_Value	[20-31]	Maximum value for this feature control
1404h	STROBE_1_INQ	Same definition as Strobe_0_Inq		
1408h	STROBE_2_INQ	Same definition as Strobe_0_Inq		
140Ch	STROBE_3_INQ	Same definition as Strobe_0_Inq		
1410h : 14Ch	Reserved			
1500h	STROBE_0_CNT	Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
			[1-5]	Reserved
		On_Off	[6]	Write: ON or OFF this function Read: read a status 0: OFF, 1: ON If this bit = 0, other fields will be read only.
		Signal_Polarity	[7]	Select signal polarity If Polarity_Inq is “1”: - Write to change strobe output polarity - Read to get strobe output polarity If Polarity_Inq is “0”: - 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.
1504h	STROBE_1_CNT	Same definition as Strobe_0_Cnt		
1508h	STROBE_2_CNT	Same definition as Strobe_0_Cnt		
150Ch	STROBE_3_CNT	Same definition as Strobe_0_Cnt		
1510h-15FFh	Reserved			

POINT GREY

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: N/A 1: Available
	[1-18]	Reserved
Count_Period	[19-23]	Controls the period of the strobe pattern Valid values: 1..16
	[24-27]	Reserved
Current_Count	[28-31]	Read-only The value of the bit index defined in GPIO_x_STRPAT_MASK that will be used during the next image's strobe. *Current_Count* increments at the same time as the strobe start signal occurs.

GPIO_STRPAT_MASK_PIN_0: 1118h

This register defines the actual strobe pattern to be implemented by GPIO0 in conjunction with the *Count_Period* defined in GPIO_STRPAT_CTRL register 110Ch (page 70).

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).

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 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

GPIO_STRPAT_MASK_PIN_1: 1128h

This register defines the actual strobe pattern to be implemented by GPIO1 in conjunction with the *Count_Period* defined in GPIO_STRPAT_CTRL register 110Ch (page 70).

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature

Field	Bit	Description
		0: N/A 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

GPIO_STRPAT_MASK_PIN_2: 1138h

This register defines the actual strobe pattern to be implemented by GPIO2 in conjunction with the *Count_Period* defined in GPIO_STRPAT_CTRL register 110Ch (page 70).

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 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

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

5.6.2 Serial Communication Using GPIO

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-232 signal levels are required, a level converter must be used to convert the TTL digital logic levels to RS-232 voltage levels. B&B Electronics (http://www.bb-elec.com/) part number 232LPTTL can be used for this conversion.

5.6.2.1 SIO Buffers

- Both the transmit and receive buffers are implemented as circular buffers that may exceed the 255 byte maximum specified by the *Buffer_Size_Inq* [24..31] field of the SERIAL_

MODE_REG register 70000h (page 75).
- 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 quadlet aligned or divisible by 4.

5.6.2.2 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 (page 75), read the available data space of the current transmit buffer *TBUF_ST* field.
2. Write characters to the SIO_DATA_REGISTER 70100h (page 75).
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.

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*
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 cam-	Set Register	0x70100	0x31323334 • *ASCII = 1234*

Step	Action	Register	Input / Expected Output
era.			
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 quadlet 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 page 71. 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.

5.6.2.3 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 (page 75), 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 (page 75).
4. To input more characters, repeat step 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*on page 72			
2. Enable the serial input (receive).	Set Register	0x70004	0x80000000
3. Verify no receive data framing errors.	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 Hyper-Terminal window.			By default, characters will not be displayed in the Hyper-Terminal 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 quadlet 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 71

5.6.2.4 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.

5.6.2.5 SIO Control and Inquiry Registers

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

Inquiry Register for SIO CSR Offset Addresses

The following register indicates the locations of the SIO CSR registers that are implemented by Point Grey IEEE-1394 cameras. These offsets are relative to the 1394 base offset 0xFFFF F0F0 0000.

Offset	Name	Field	Bit	Description
488h	SIO_CONTROL_ CSR_INQ	SIO_Control_ Quadlet_Offset	[0-31]	Quadlet offset of the SIO control CSRs from the base address of initial register space

Current SIO Register Offsets

At the time of this revision, the SIO offsets that would be derived using the quadlet offset information in *Inquiry Registers for SIO CSR Offset Addresses* (page 75) are as follows:

The following table of SIO offsets is current as of the revision date. These offsets are subject to change without notice. Refer to Calculating Register Adresses using Quadlet Offsets on page 126 for information on calculating the actual offsets of the following registers.

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

Offset	Name	Field	Bit	Description
70000h	SERIAL_ MODE_REG	Baud_ Rate	[0-7]	*Baud rate setting* Write: Set baud rate Read: Get current baud rate 0: 300bps 1: 600bps 2: 1200bps 3: 2400bps 4: 4800bps 5: 9600bps

Offset	Name	Field	Bit	Description
				6: 19200bps 7: 38400bps 8: 57600bps 9: 115200bps 10: 230400bp Other values reserved
		Char_ Length	[8-15]	Character length setting Write: Set data length (must not be 0) Read: Get data length 7: 7bits 8: 8bits Other values reserved
		Parity	[16-17]	*Parity setting* Write: Set parity Read: Get current parity 0: None 1: Odd 2: Even
		Stop_ Bit	[18-19]	*Stop bits* Write: Set stop bit Read: Get current 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 Bufferson page 71. If this value=1, *Buffer_Status_Control* and *SIO_ Data_Register* characters 1-3 should be ignored.
70004h	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

Offset	Name	Field	Bit	Description
		-	[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
70008h	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
7000Ch	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

Offset	Name	Field	Bit	Description
				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
70010h : 700FFh	Reserved			
70100h	SIO_DATA_ REGISTER	Char_0	[0-7]	*Character_0* Read: Read character from receive buffer. Padding data if data is not available. Write: Write character to transmit buffer. Padding data if data is invalid.
		Char_1	[8-16]	*Character_1* Read: Read character from receive buffer+1. Padding data if data is not available. Write: Write character to transmit buffer+1. Padding data if data is invalid.
		Char_2	[17-23]	*Character_2* Read: Read character from receive buffer+2. Padding data if data is not available. Write: Write character to transmit buffer+2. Padding data if data is invalid.
		Char_3	[24-31]	*Character_3* Read: Read character from receive buffer+3. Padding data if data is not available. Write: Write character to transmit buffer+3. Padding data if data is invalid.
70104h : 701FFh	SIO_DATA_ REGISTER_ ALIAS		[0-31]	Alias SIO_Data_Register area for block transfer.

5.6.3 Pulse Width Modulation (PWM)

When the *Pin_Mode* field of GPIO_CTRL_PIN_x register (page 79) is set to GPIO_MODE_4, pin x 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 PWM's), but the pulse can appear on any of the GPIO pins.

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

The pin configured to output a PWM signal (PWM pin) remains in the same state at the time the 'enable pin' is disabled. For example, if the PWM is in a high signal state when the 'enable pin' is disabled, the PWM pin remains in a high state. To re-set the pin signal, you must re-configure the PWM pin from GPIO_Mode_4 to GPIO_Mode_1.

To configure the camera to generate an infinite number of PWM pulses, set the *Pwm_Count* to 0xFF (255). To stop the infinite PWM pulse mode, set the *Pwm_Count* to 0x00 (0), or take the GPIO pin out of PWM mode by setting *Pin_Mode* to 0x00 (0).

Format of GPIO_CTRL_PIN_x Register in GPIO_Mode_4

Field	Bit	Description
Presence_Inq	[0]	Value should be '1'
	[1-11]	Reserved
Pin_Mode	[12-15]	Value should be '4'
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. For example, if this bit is zero, the PWM will be 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

GPIO_CTRL_PIN_0: 1110h

This register provides control over the first GPIO pin (Pin 0).

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-11]	Reserved
Pin_Mode	[12-15]	Current GPIO_Mode 0: Input 1: Output 4: Pulse width modulation (PWM)
Data	[16-31]	Data field GPIO_MODE_0 – bit 31 contains value GPIO_MODE_1 – bit 31 contains value GPIO_MODE_4 – see *Pulse Width Modulation* on page 78

GPIO_XTRA_PIN_0: 1114h

This register contains mode specific data for the first GPIO pin (Pin 0). Units are ticks of a 1.024MHz clock.

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)

GPIO_CTRL_PIN_1: 1120h

This register provides control over the second GPIO pin (Pin 1).

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-11]	Reserved
Pin_Mode	[12-15]	Current GPIO_Mode 0: Input 1: Output 4: Pulse width modulation (PWM)
Data	[16-31]	Data field GPIO_MODE_0 – bit 31 contains value GPIO_MODE_1 – bit 31 contains value GPIO_MODE_4 –see *Pulse Width Modulation*on page 78

GPIO_XTRA_PIN_1: 1124h

This register contains mode specific data for the second GPIO pin (Pin 1). Units are ticks of a 1.024MHz clock.

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)

GPIO_CTRL_PIN_2: 1130h

This register provides control over the third GPIO pin.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-11]	Reserved
Pin_Mode	[12-15]	Current GPIO_Mode 0: Input 1: Output 4: Pulse width modulation (PWM)
Data	[16-31]	Data field GPIO_MODE_0 – bit 31 contains value GPIO_MODE_1 – bit 31 contains value GPIO_MODE_4 – see *Pulse Width Modulation*on page 78

GPIO_XTRA_PIN_2: 1134h

This register contains mode specific data for the third GPIO pin. Units are ticks of a 1.024MHz clock.

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)

GPIO_CTRL_PIN_3: 1140h

This register provides control over the fourth GPIO pin.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-11]	Reserved
Pin_Mode	[12-15]	Current GPIO_Mode 0: Input 1: Output 4: Pulse width modulation (PWM)
Data	[16-31]	Data field GPIO_MODE_0 – bit 31 contains value

Field	Bit	Description
		GPIO_MODE_1 – bit 31 contains value GPIO_MODE_4 – see *Pulse Width Modulation*on page 78

GPIO_XTRA_PIN_3: 1144h

This register contains mode specific data for the fourth GPIO pin. Units are ticks of a 1.024MHz clock.

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)

5.7 On-Camera Frame Buffer

The camera has 32MB 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 in the frame buffer by enabling the *Image_Buffer_Ctrl* bit of IMAGE_RETRANSMIT register 634h (page 83) . 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 in which the camera will acquire 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 *Image_Buffer_Ctrl* bit of register 634h. Enabling this bit 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, the *ISO_EN / Continuous Shot* field of register 614h must be disabled, and the *Transfer_Data_Select* field of register 634h set to 1.
5. **Transmit images off of the camera**
   The final step involves poking the ONE_SHOT/MULTI_SHOT register 61Ch 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 flash memory (page 27). Accessing flash memory is significantly slower than accessing the frame buffer, and storage is limited.

IMAGE_RETRANSMIT: 634h

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

Transmitting buffered data is available when ISO_EN = 0. (See ISO_EN / CONTINUOUS_SHOT: 614h on page 45.) Either One_shot or Multi_shot can be used to transmit buffered data when *Transfer_Data_Select* = 1. (See ONE_SHOT / MULTI_SHOT: 61Ch on page 45.) 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

Field	Bit	Description
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.

5.7.1 Retransmitting an Image in External Trigger Mode

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.

5.7.2 Storing images for later transmission

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.

5.8 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 hue, saturation, sharpness, white balance or gamma.

Enabling raw Bayer output when operating in a monochrome data format (page 48) produces an image shift effect in the test pattern.

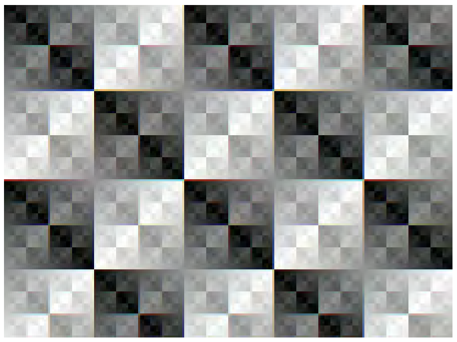


Figure 5.12: Test pattern sample image

TEST_PATTERN: 104Ch

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A, 1: Available
	[1-30]	Reserved
Test_Pattern_1	[31]	Value 0: Disable test pattern 1: Enable test pattern

6 Imaging Parameters and Control

6.1 Brightness

The camera supports manual brightness control using BRIGHTNESS register 800h (page 87). Brightness, also known as offset, controls the level of black in an image.

BRIGHTNESS: 800h

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 the Absolute Value CSRs section of the Appendix (page 151).

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A, 1: Available
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.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Write: 1: Begin to work (self-cleared after operation) Read: 0: Not in operation, 1: In operation If A_M_Mode = 1, this bit is ignored
ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Write: set the mode Read: read a current mode 0: Manual, 1: Automatic
	[8-19]	Reserved
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

6.1.1 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 );
```

6.2 Shutter

The camera supports automatic, manual and one-push control of the CCD shutter time. Refer to the *Specifications* section for supported shutter time ranges. 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 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 (see *Camera Power*); **and**
2. The camera is not in 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 PC (See ISO_EN / CONTINUOUS_SHOT: 614h on page 45.). 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 43.

6.2.1 Extended Shutter Times

The maximum shutter time can be extended beyond the normal shutter range by setting the *ON_OFF* bit [6] of the FRAME_RATE register 0x83C (page 38) to zero (OFF). Once the FRAME_RATE is turned off, you should see the *Max_Value* of the ABS_VAL_SHUTTER register (page 152) increase.

The longest possible shutter times are available only when operating the camera in Format_7 Mode_3 or Format_7 Mode_8 (page 37).

The maximum extended shutter time reported by the SHUTTER_INQ register 51Ch (page 140) 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 152) to determine the maximum shutter.

Related Knowledge Base Articles

ID	Title	URL
166	Extended shutter mode operation for IIDC-compliant Point Grey Imaging Products.	www.ptgrey.com/support/kb/index.asp?a=4&q=166

Related Resources

Title	Link
FlyCapture SDK *ExtendedShutterEx* sample program	ExtendedShutterEx

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 152). 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 92). Both the *Value* field and the ABS__VAL_SHUTTER register become read only.

Converting fixed-point values to absolute values

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
...	...	...

The chart above can be expressed in terms of the following formula:

If fixed value <= 1024:
absolute_value = 10 us * *fixed_value*

Else:
absolute_value = 10 us * (512 + *fixed_value* % 512) * (2 ** ((*fixed_value*/512) - 1))

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A, 1: Available
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 ignored.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Write: 1: Begin to work (self-cleared after operation) Read: 0: Not in operation, 1: In operation If A_M_Mode = 1, this bit is ignored
ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Write: set the mode Read: read a current 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.

6.2.2 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 );
```

6.3 Gain

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. On ZBR2-GEHD-50S5C models, gain is configurable in steps of 0.36 dB.

For information about timing changes to the gain setting, see *When Camera Property Settings Take Effect* on page 43.

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 88) first.

GAIN: 820h

This register has three states:

State	Description
Manual/Abs	The gain value is set by the user via the ABS_VAL_GAIN register (page 152). The *Value* field becomes read only and reflects the converted absolute value.
Manual	The gain value is set by the *Value* field: the ABS_VAL_GAIN register becomes read only and contains the current gain.
Auto	The gain value is set by the auto exposure controller (if enabled) (page 92): both the *Value* field and the ABS_VAL_GAIN 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 the Absolute Value CSRs section of the Appendix (page 151).

Format:

Same format as BRIGHTNESS: 800h on page 87

6.3.1 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 );
```

6.4 Auto Exposure

Auto exposure (AE) 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 using the AUTO_EXPOSURE_RANGE (page 93), AUTO_SHUTTER_RANGE (page 94) and AUTO_GAIN_RANGE (page 94) registers.

The AUTO_EXPOSURE register 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 the SHUTTER (page 89) and GAIN (page 91) registers. This mode is achieved by setting the *ON_OFF* field to be 0. An equivalent mode can be achieved by setting the *A_M_Mode* fields in the SHUTTER and GAIN registers to 0 (Manual).
ON Manual Exposure Control	The camera automatically modifies the SHUTTER and GAIN registers to try and match the average image intensity to the value written to the *Value* field. This mode is achieved by setting the *A_M_Mode* value of the AUTO_ EXPOSURE register to 0 (manual) and either/both of the *A_M_Mode* values for the SHUTTER and GAIN registers to 1 (Auto).
ON Auto Exposure Control	The camera modifies the *Value* field in order to produce an image that is visually pleasing. This mode is achieved by setting the *A_M_MODE* for all three of the AUTO_EXPOSURE, SHUTTER and GAIN registers to 1 (Auto). In this mode, the *Value* field reflects the average image intensity.

Auto exposure can only control the exposure when the SHUTTER and/or GAIN registers have their *A_M_Mode* bits set. If only one of the registers is in "auto" mode then the auto exposure controller attempts to control the image intensity using just that one register. If both of these registers 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).

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 the Absolute Value CSRs section of the Appendix (page 151).

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

AUTO_EXPOSURE: 804h

Format:

Same format as SHUTTER: 81Ch on page 89.

AUTO_EXPOSURE_RANGE: 1088h

Specifies the range of allowed exposure values to be used by the automatic exposure controller when in auto mode. Fixed point (relative) values must be specified. Do not specify absolute values.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature

Field	Bit	Description
		0: N/A 1: Available
	[1-7]	Reserved
Min_Value	[8-19]	Lower bound
Max_Value	[20-31]	Upper bound

AUTO_SHUTTER_RANGE: 1098h

Allows the user to specify the range of shutter values to be used by the automatic exposure controller - generally some subset of the entire shutter range described by SHUTTER_INQ register 51Ch (page 140). Fixed point (relative) values must be specified. Do not specify absolute values.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 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 on page 89 for the list of ranges).

AUTO_GAIN_RANGE: 10A0h

Allows the user to specify the range of gain values to be used by the automatic exposure controller - generally some subset of the entire gain range described by GAIN_INQ register 520h (page 140).

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-5]	Reserved
ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF 1: ON If this bit = 0, other fields will be read only

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

AE_ROI: 1A70 – 1A74h

This register 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 will be used.

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]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields will be read only
			[7-31]	Reserved
1A74h	AE_ROI_OFFSET		[0-31]	Quadlet offset of the base address for the ROI CSR's
Base+0h (1C40h)	AE_ROI_UNIT_POSITION_INQ	Hposunit	[0-15]	Horizontal units for position
		Vposunit	[16-31]	Vertical units for position
Base+4h (1C44h)	AE_ROI_UNIT_SIZE_INQ	Hunit	[0-15]	Horizontal units for size
		Vunit	[16-31]	Vertical units for size
Base+8h (1C48h)	AE_ROI_POSITION	Left	[0-15]	Left position of ROI
		Top	[16-31]	Top position of ROI
Base+Ch (1C4Ch)	AE_ROI_SIZE	Width	[0-15]	Width of ROI
		Height	[16-31]	Height of ROI

6.4.1 Example Setting Auto Exposure Using the FlyCapture2 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 );
```

6.5 Gamma and Lookup Table

The camera supports gamma and lookup table (LUT) functionality. CCD manufacturers strive to make the transfer characteristics of CCDs 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 controlled using the GAMMA register 0x818 (page 96). Gamma 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, set gamma to 1, or turn gamma off.

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

GAMMA: 818h

This register provides a mechanism to control the function used to non-linearly map a higher bit depth image produced by the sensor to the requested number of bits.

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 the Absolute Value CSRs section of the Appendix (page 151).

Format:

Same format as BRIGHTNESS: 800h on page 87.

When ON is written to the ON_OFF field, the ON_OFF field of the LUT register (page 97) is turned to OFF.

LUT: 80000h – 80048h

This register allows the user to access and control a lookup table (LUT), with entries stored on-board the camera. The LUT will also be modifed under the following circumstances:

- Camera reinitialization via the INITIALIZE register 000h
- Changing the CURRENT_VIDEO_MODE or CURRENT_VIDEO_FORMAT registers 604h or 608h
- Changing the GAMMA register 818h or ABS_VAL_GAMMA register

The look up table function 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^{Input_Depth} entries where each entry is *Output_ Depth* bits wide. Channel table entries shall be padded to 32 bits or a complete quadlet.

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.

Offset	Name	Field	Bit	Description
80000h	LUT_Ctrl_ Inq (Read Only)	Presence_ Inq	[0]	Presence of this feature 0:N/A 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_	Read_Bank_	[0]	Capability of reading data from Bank 0

Offset	Name	Field	Bit	Description
	Rd_Inq	0_Inq		
		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
	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

Offset	Name	Field	Bit	Description
		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: N/A 1: Available
			[1..4]	Reserved
		ON_OFF	[5]	Write: ON or OFF this feature Read: read a status 0: OFF 1: ON When ON is written, the ON_OFF field of the GAMMA register (page 96) is turned to OFF.
			[6..27]	Reserved
		Active_Bank	[28..31]	Active bank
8000Ch	Bank_0_Offset_Inq	Bank_0_Quadlet_Offset	[0..31]	Quadlet offset of Bank 0 table data
80010h	Bank_1_Offset_Inq	Bank_1_Quadlet_Offset	[0..31]	Quadlet offset of Bank 1 table data
80014h	Bank_2_Offset_Inq	Bank_2_Quadlet_Offset	[0..31]	Quadlet offset of Bank 2 table data
80018h	Bank_3_Offset_Inq	Bank_3_Quadlet_Offset	[0..31]	Quadlet offset of Bank 3 table data
8001Ch	Bank_4_Offset_Inq	Bank_4_Quadlet_Offset	[0..31]	Quadlet offset of Bank 4 table data

Offset	Name	Field	Bit	Description
80020h	Bank_5_Offset_Inq	Bank_5_Quadlet_Offset	[0..31]	Quadlet offset of Bank 5 table data
80024h	Bank_6_Offset_Inq	Bank_6_Quadlet_Offset	[0..31]	Quadlet offset of Bank 6 table data
80028h	Bank_7_Offset_Inq	Bank_7_Quadlet_Offset	[0..31]	Quadlet offset of Bank 7 table data
8002Ch	Bank_8_Offset_Inq	Bank_8_Quadlet_Offset	[0..31]	Quadlet offset of Bank 8 table data
80030h	Bank_9_Offset_Inq	Bank_9_Quadlet_Offset	[0..31]	Quadlet offset of Bank 9 table data
80034h	Bank_10_Offset_Inq	Bank_10_Quadlet_Offset	[0..31]	Quadlet offset of Bank 10 table data
80038h	Bank_11_Offset_Inq	Bank_11_Quadlet_Offset	[0..31]	Quadlet offset of Bank 11 table data
8003Ch	Bank_12_Offset_Inq	Bank_12_Quadlet_Offset	[0..31]	Quadlet offset of Bank 12 table data
80040h	Bank_13_Offset_Inq	Bank_13_Quadlet_Offset	[0..31]	Quadlet offset of Bank 13 table data
80044h	Bank_14_Offset_Inq	Bank_14_Quadlet_Offset	[0..31]	Quadlet offset of Bank 14 table data
80048h	Bank_15_Offset_Inq	Bank_15_Quadlet_Offset	[0..31]	Quadlet offset of Bank 15 table data

Lookup Table Data Structure

Each bank of channels is composed of entries padded to a complete 32-bit quadlet. 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) … …

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

6.5.1 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 );
```

6.6 Saturation

The camera supports saturation adjustment of color values.

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

<u>SATURATION: 814h</u>

This register 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.

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 the Absolute Value CSRs section of the Appendix (page 151).

Format:

Same format as BRIGHTNESS: 800h on page 87.

6.6.1 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 );
```

6.7 Hue

The camera supports hue adjustment, which defines the color phase of images.

HUE: 810h

This register 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.

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 the Absolute Value CSRs section of the Appendix (page 151).

Format:

Same format as BRIGHTNESS: 800h on page 87.

6.7.1 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 );
```

6.8 Sharpness

The camera supports sharpness adustment, 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.

SHARPNESS: 808h

Format:

Same format as BRIGHTNESS: 800h on page 87.

6.8.1 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 );
```

6.9 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 white balance scheme outlined in the IIDC specification states that blue and red are adjustable and that green is not. The blue and red values can be controlled using the WHITE_BALANCE register 0x80C (page 105).

The camera also implements 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.

White balance may be unresponsive in auto mode if auto exposure (page 93) is < 0.1 EV (approximately).

WHITE_BALANCE: 80Ch

This register controls the relative gain of pixels in the Bayer tiling used in the CCD of a color camera. Control of the register is achieved via the *R_Value* and *B_Value* fields and the *On_Off* bit. Both value fields specify relative gain, with a value that is half the maximum value being a relative gain of zero. This register has two states:

- *OFF - the same gain is applied to all pixels in the Bayer tiling.*
- *ON - the R_Value field is applied to the red pixels of the Bayer tiling and the B_Value field 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

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with value in the Value field 1: Control with value in the associated Abs 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) Write: 1: Begin to work (self-cleared after operation) Read: 0: Not in operation, 1: In operation

Field	Bit	Description
		If A_M_Mode = 1, this bit is ignored
ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Write: Set the mode. Read: read the current 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.

6.9.1 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 );
```

6.10 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 108).

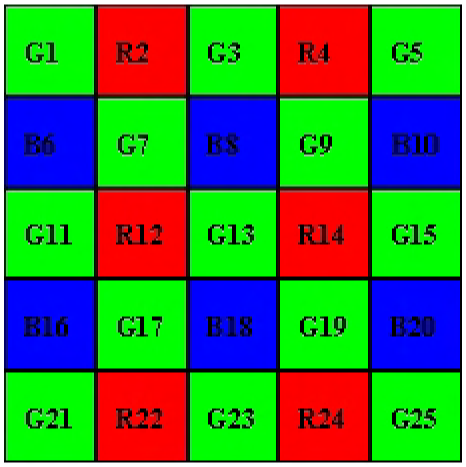


Figure 6.1: Example Bayer tile pattern

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 CCD.

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:

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

6.10.1 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 Video Formats, Modes and Frame Rates on page 35). Alternatively, use the BAYER_MONO_CTRL register 1050h (page 48). This register enables raw Bayer output in non-Format_7 Y8 / Y16 modes, or Format_7 Mono8 / Mono16 modes.

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. This format can be queried using the BAYER_TILE_MAPPING register 0x1040 (page 108).

Using the FlyCapture 2.0 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)

Related Knowledge Base Articles

ID	Title	URL
33	Different color processing algorithms.	www.ptgrey.com/support/kb/index.asp?a=4&q=33
37	Writing color processing software and color interpolation algorithms.	www.ptgrey.com/support/kb/index.asp?a=4&q=37

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.

Field	Bit	Description
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.

6.11 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 82).

MIRROR_IMAGE_CTRL: 1054h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature. 0: N/A, 1: Available
	[1-30]	Reserved.
Mirror_Image_Ctrl	[31]	Value 0: Disable horizontal (mirror) image flip 1: Enable horizontal (mirror) image flip

6.12 Embedded Image Information

The camera provides a feature that allows image timestamps, region of interest (ROI) and other camera settings information to be embedded in the first several pixels of each image.

FRAME_INFO: 12F8h

This register 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 PC 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 1 quadlet (4 bytes) of the image. When the camera is operating in Y8 (8bits/pixel) mode, this is therefore 4 pixels worth of data.

Insertion of each quadlet is controlled by a bit in this register. Because it is a bit field, quadlets appear in reverse order from the bits that control them. So, setting bit 31 to ‘1’ enables Timestamp embedding, bit 30 enables Gain, etc. For black and white cameras, white balance is still included, but no valid data is provided.

For example, a write of 800003FFh to this register would turn on all possible options. Therefore, the first 10 quadlets (40 bytes) of image data would contain camera information in the following format, when accessed using the FlyCapture 2.0 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 (0x12F8 = 0x80000004), 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.

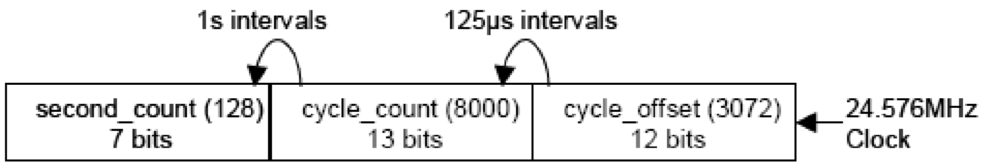
Format:

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

Field	Bit	Description	Frame-Specific Information
Insert_Info	22	Display image-specific information 0: Off 1: On	Region of Interest (ROI) position (See “Interpreting ROI Information” below)
	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 “Interpreting Timestamp information” below)

Interpreting Timestamp information

The Timestamp format matches the CYCLE_TIME 0xFF100200 register format 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 of the quadlet 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.

6.13 Pixel Defect Correction

Point Grey tests for white blemish pixels on each camera that is produced. 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. The maximum number of pixels corrected is 60.

Related Knowledge Base Articles

ID	Title	URL
314	How Point Grey tests for white blemish pixels	www.ptgrey.com/support/kb/index.asp?a=4&q=314

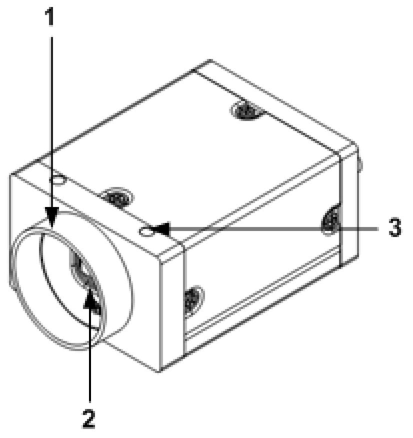
PIXEL_DEFECT_CTRL: 1A60h

Format:

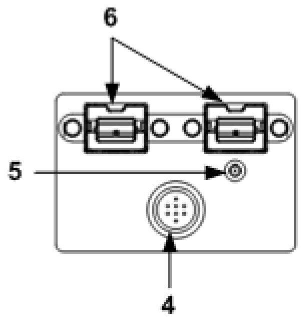
Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-5]	Reserved
ON_OFF	[6]	Enable or disable pixel correction 0: Off 1: On
Reserved	[7]	Reserved
Max_Pixels	[8-19]	Maximum number of pixels that can be corrected
Cur_Pixels	[20-31]	Current number of pixels that are being corrected

7 Camera Mechanical Properties

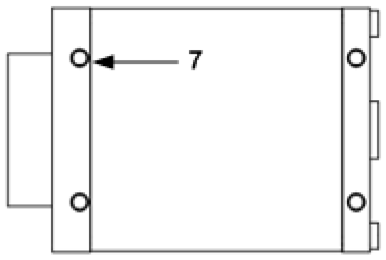
7.1 Physical Description



Front/Side/Top View



Rear View



Bottom View

1. Lens holder (C-mount) (page 115)
2. Glass / IR filter system (page 117)
3. M3 x 0.5 mm mounting holes (page 117)
4. General Purpose I/O Connector (page 30)
5. Status LED (page 120)
6. IEEE-1394 connectors (page 29)
7. M3 x 0.5 mm mounting holes (page 117)

7.2 Camera Dimensions

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

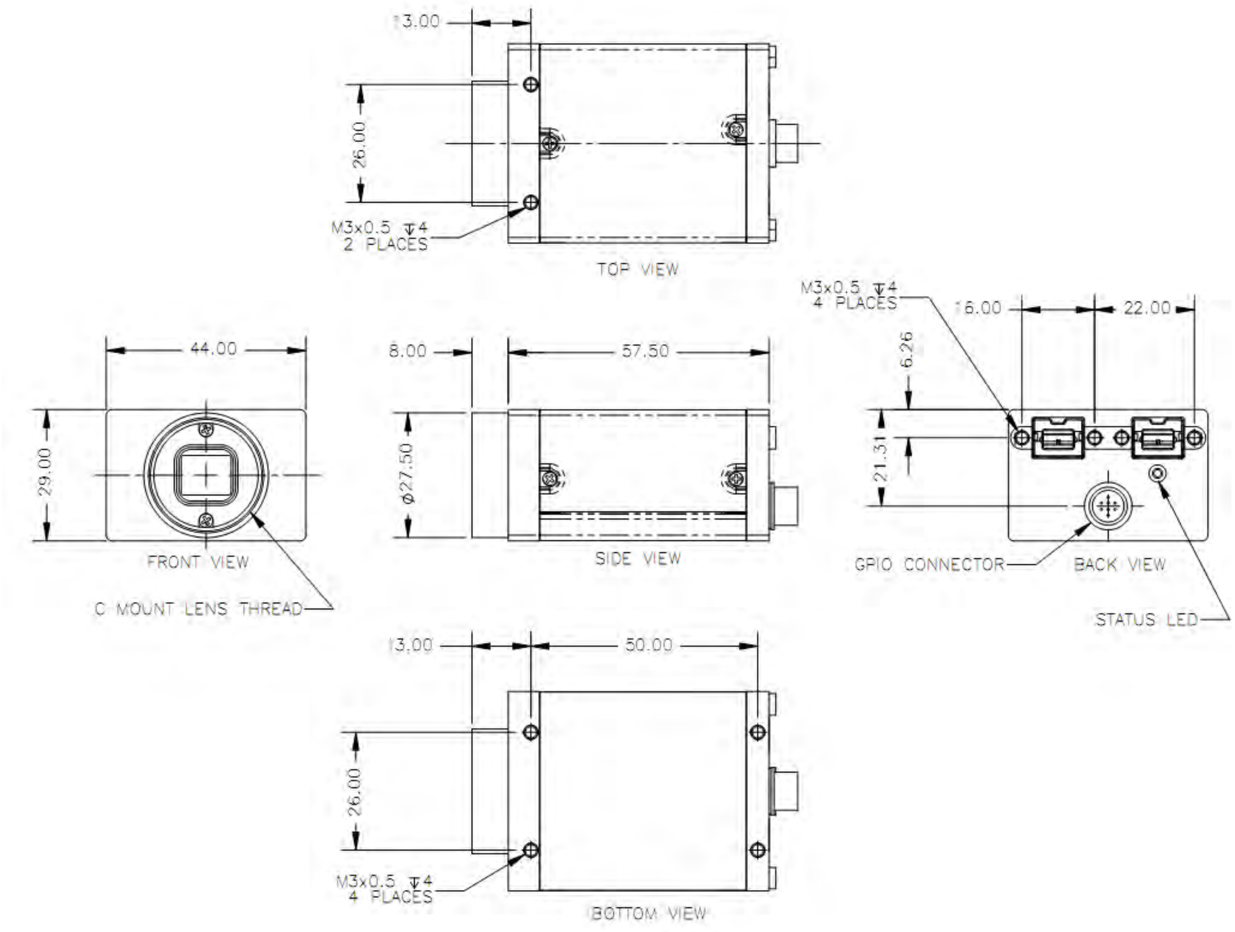


Figure 7.1: Camera Dimensional Diagram

7.3 Tripod Adapter Dimensions

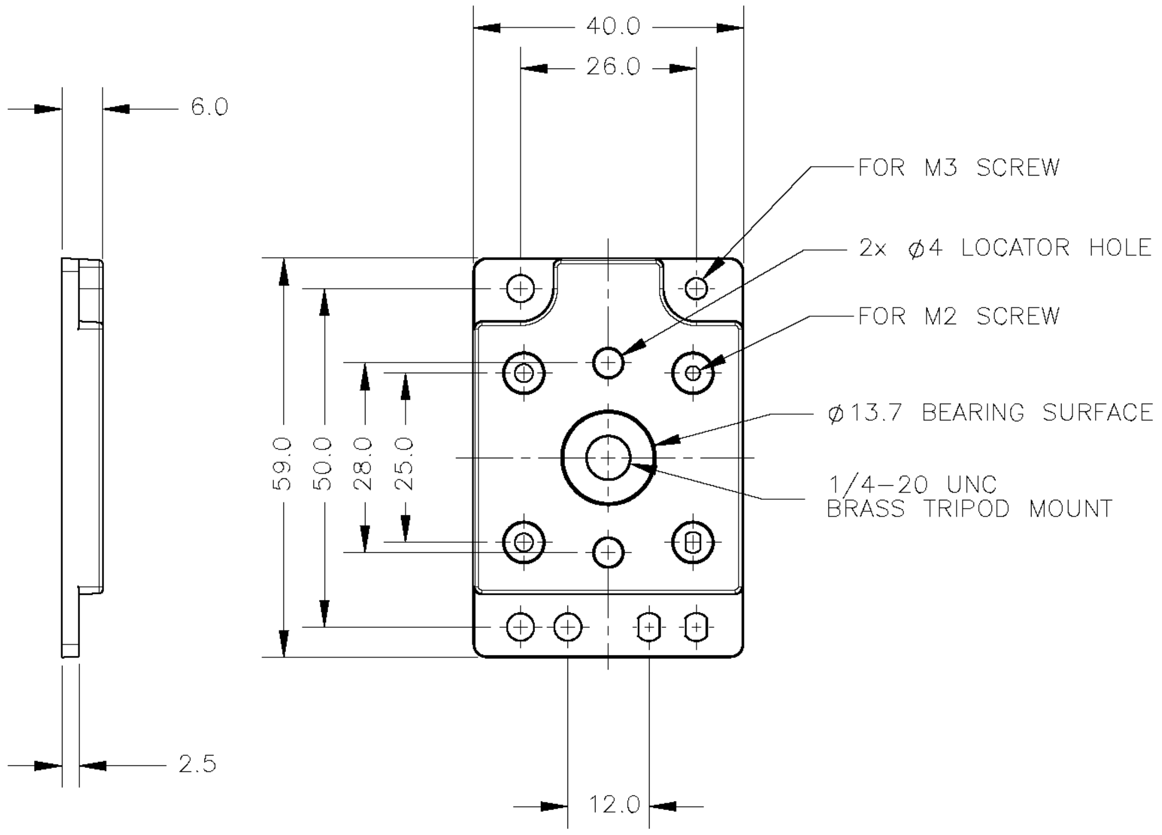
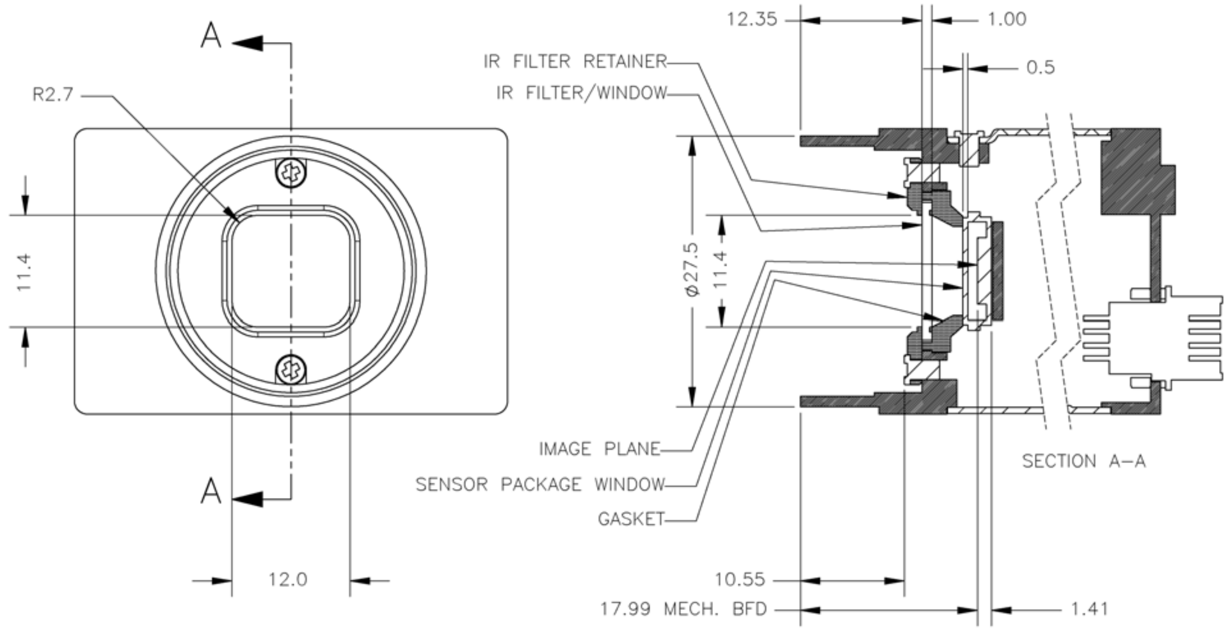


Figure 7.2: Tripod Adapter Dimensional Diagram

7.4 Lens Mount Dimensions

The lens holder is compatible with C-mount lenses. Lenses are not included with individual cameras.



Although the C-mount lens specification for back flange distance (BFD) is 17.52 mm, these distances are offset due to the presence of both a 1 mm infrared cutoff (IRC) filter and a 0.5 mm sensor package window. These two pieces of glass fit between the lens and the sensor image plane. The IRC filter is installed on color cameras. In monochrome cameras, it is a transparent piece of glass. The sensor package window is installed by the sensor manufacturer. Both components cause refraction, which requires some offset in flange back distance to correct. The resulting BFD is 17.99 mm.

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

7.5 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 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 described at the beginning of this reference manual.*
- *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

ID	Title	URL
215	Removing the IR filter from a color camera	http://www.ptgrey.com/support/kb/index.asp?a=4&q=215
345	Selecting a lens for your camera	http://www.ptgrey.com/support/kb/index.asp?a=4&q=345

7.6 Mounting

7.6.1 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

7.6.2 Using the Tripod Adapter

The tripod mounting bracket is equipped with four (4) M3 mounting holes. For more information, see Tripod Adapter Dimensions on page 1.

7.7 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.

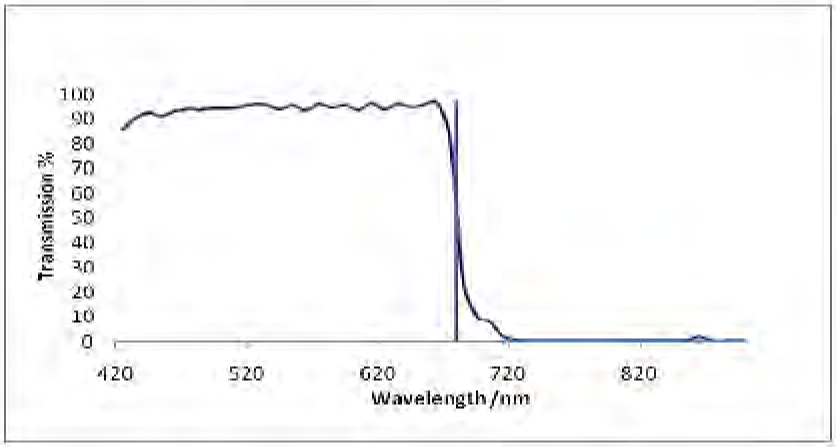


Figure 7.3: IR filter transmittance graph

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 or BK7 equivalent for coating filters
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 page 116.

Related Knowledge Base Articles

ID	Title	URL
215	Removing the IR filter from a color camera	http://www.ptgrey.com/support/kb/index.asp?a=4&q=215

8 Troubleshooting and Support

8.1 Technical Support Resources

Point Grey Research Inc. endeavors to provide the highest level of technical support possible to our customers. Most support resources can be accessed through the Product Support section of our website: www.ptgrey.com/support.

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, e-mail address, and camera serial number. To apply for a Customer Login Account go to www.ptgrey.com/support/downloads/.

Knowledge Base

Our on-line knowledge base at www.ptgrey.com/support/kb/ 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 downloads site at www.ptgrey.com/support/downloads. 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 at www.ptgrey.com/support/contact/.

8.1.1 Status Indicator LED

LED Status	Description
Maximum red (Initial connection)	Initial startup. On until camera is initialized.
Maximum red (During operation)	Condition 1: Bus Rest. On for 0.66s. Condition 2: Power failure. On until power-up via CAMERA_POWER 0x610 (page 16).
Dull Red	Configuration error.
Bright Red	Configuration error.
Dull Green	Camera is idle.
Bright Green	Firewire activity. On for 0.5s during activity.
Dull Yellow	Powered down.
Bright Yellow	Powered down + activity. On for 0.5s during activity.
Red/Green flashing	Camera firmware is being updated. Flashes at 5Hz.

Table 8.1: Status indicator LED descriptions

LED_CTRL: 1A14h

This register allows the user to turn off the camera's status LED. LED(s) are re-enabled the next time the camera is power cycled.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
	[1-22]	Reserved
LED_Ctrl	[23-31]	Enable or disable the LED 0x00: Off, 0x74: On

8.2 Common Image Sensor Artifacts

White Blemish Pixels

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. For more information, see *Blemish Pixel Defect Correction* (page 111).

Vertical Smear

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. More information and suggestions for troubleshooting are presented in Knowledge Base Article 88 (http://www.ptgrey.com/support/kb/index.asp?a=4&q=88)

Dead / Hot Pixels

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

8.3 Error Status Registers

VMODE_ERROR_STATUS: 628h

This register is used by the camera to report any camera configuration errors. If an error has occurred, no image data will be sent by the camera.

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.

XMIT_FAILURE: 12FCh

This register 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.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: N/A 1: Available
Frame_Count	[1-31]	Read: Count of failed frame transmissions. Write: Reset.

CAMERA_LOG: 1D00 – 1DFFh

This register provides access to the camera's 256 byte internal message log, which is often useful for debugging camera problems. Characters are hexadecimal representations of ASCII characters. Contact technical support for interpretation of message logs.

Format:

Offset	Description
1D00..1DFF	Each byte is the hexadecimal representation of an ASCII

Offset	Description
	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.

Appendix A: General Register Information

Register Memory Map

The IEEE-1394 specification 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	1394 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);
```

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

```
cam.ReadRegister(0x81C, &regVal);
```

Config ROM

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

Unit Directory

	Offset	Bit	Description
Unit Directory	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

Unit Dependent Info

	Offset	Bit	Description
Unit Dependent Info	0000h	[0-15]	unit_dep_info_length
		[16-31]	CRC
	0004h	[0-7]	40h
		[8-31]	command_regs_base
	0008h	[0-7]	81h
		[8-31]	vendor_name_leaf
	000Ch	[0-7]	82h
		[8-31]	model_name_leaf
	0010h	[0-7]	38h
		[8-31]	unit_sub_sw_version
	0014h	[0-7]	39h
		[8-31]	Reserved
	0018h	[0-7]	3Ah
		[8-31]	Reserved
	001Ch	[0-7]	3Bh
		[8-31]	Reserved
	0020h	[0-7]	3Ch
		[8-31]	vendor_unique_info_0
	0024h	[0-7]	3Dh
		[8-31]	vendor_unique_info_1
	0028h	[0-7]	3Eh
		[8-31]	vendor_unique_info_2
	002Ch	[0-7]	3Fh
		[8-31]	vendor_unique_info_3

- *command_regs*_base is the quadlet offset from the base address of initial register space of the base address of the command registers.
- *vendor_name_leaf* specifies the number of quadlets 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.
- *model_name_leaf* specifies the number of quadlets 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.
- *unit_sub_sw_version* specifies 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

Calculating Register Addresses using Quadlet Offsets

The addresses for many IIDC control and status registers (CSR's), such as those that provide control over absolute values, Format_7 video modes, PIO, SIO and strobe output, vary between camera manufacturers. In order to provide a common mechanism across camera models for determining the location of these CSR's relative to the 1394 base address, the IIDC specification provides fixed locations for inquiry registers that contain quadlet offsets, or pointers, to the actual offsets.

For example, the Absolute Value CSR's provide minimum, maximum and current real-world values for camera properties such as gain, shutter, etc., as described in *Absolute Value CSR Registers*on page 151. 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 quadlet offset for the absolute value CSR for shutter.

    ```
    unsigned int ulValue;

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

2. The 32-bit ulValue is a quadlet 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;
    ```

Appendix B: Inquiry Registers

Inquiry Registers for Video Format / Mode / Frame Rate

The following registers may be used to determine the video formats, modes and frame rates that are available with the camera.

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

Video Format Inquiry Registers

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

Video Mode Inquiry Registers

Format:

Offset	Name	Field	Bit	Description
180h	V_MODE_INQ_O (Format 0)	Mode_0	[0]	160x120 YUV(4:4:4) Mode (24bit/pixel)
		Mode_1	[1]	320x240 YUV(4:2:2) Mode (16bit/pixel)
		Mode_2	[2]	640x480 YUV(4:1:1) Mode (12bit/pixel)
		Mode_3	[3]	640x480 YUV(4:2:2) Mode (16bit/pixel)
		Mode_4	[4]	640x480 RGB Mode (24bit/pixel)
		Mode_5	[5]	640x480 Y8 (Mono) Mode (8bit/pixel)
		Mode_6	[6]	640x480 Y16 (Mono16) Mode (16bit/pixel)
			[7-31]	Reserved
184h	V_MODE_INQ_1 (Format 1)	Mode_0	[0]	800x600 YUV(4:2:2) Mode (16bit/pixel)
		Mode_1	[1]	800x600 RGB Mode (24bit/pixel)
		Mode_2	[2]	800x600 Y (Mono) Mode (8bit/pixel)
		Mode_3	[3]	1024x768 YUV(4:2:2) Mode (16bit/pixel)
		Mode_4	[4]	1024x768 RGB Mode (24bit/pixel)
		Mode_5	[5]	1024x768 Y (Mono) Mode (8bit/pixel)
		Mode_6	[6]	800x600 Y (Mono16) Mode (16bit/pixel)
		Mode_7	[7]	1024x768 Y (Mono16) Mode (16bit/pixel)
			[8-31]	Reserved
188h	V_MODE_INQ_2 (Format 2)	Mode_0	[0]	1280x960 YUV(4:2:2) Mode (16bit/pixel)
		Mode_1	[1]	1280x960 RGB Mode (24bit/pixel)
		Mode_2	[2]	1280x960 Y (Mono) Mode (8bit/pixel)
		Mode_3	[3]	1600x1200 YUV(4:2:2) Mode (16bit/pixel)
		Mode_4	[4]	1600x1200 RGB Mode (24bit/pixel)
		Mode_5	[5]	1600x1200 Y (Mono) Mode (8bit/pixel)
		Mode_6	[6]	1280x960 Y (Mono16) Mode (16bit/pixel)
		Mode_7	[7]	1600X 1200 Y (Mono16) Mode (16bit/pixel)
			[8-31]	Reserved
18Ch : 197h	Reserved			

Offset	Name	Field	Bit	Description
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

Video Frame Rate Inquiry Registers

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

Format:

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]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
204h	V_RATE_INQ_0_1 (Format 0, Mode 1)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
208h	V_RATE_INQ_0_2 (Format 0, Mode 2)	FrameRate_0	[0]	1.875fps

Offset	Name	Field	Bit	Description
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
20Ch	V_RATE_INQ_0_3 (Format 0, Mode 3)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
210h	V_RATE_INQ_0_4 (Format 0, Mode 4)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
214h	V_RATE_INQ_0_5 (Format 0, Mode 5)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps

Offset	Name	Field	Bit	Description
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
218h	V_RATE_INQ_0_6 (Format 0, Mode 6)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
21Ch : 21Fh	Reserved			
220h	V_RATE_INQ_1_0 (Format 1, Mode 0)	FrameRate_0	[0]	Reserved
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
224h	V_RATE_INQ_1_1 (Format 1, Mode 1)	FrameRate_0	[0]	Reserved
		FrameRate_1	[1]	Reserved
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
228h	V_RATE_INQ_1_2 (Format 1, Mode 2)	FrameRate_0	[0]	Reserved

Offset	Name	Field	Bit	Description
		FrameRate_1	[1]	Reserved
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
22Ch	V_RATE_INQ_1_3 (Format 1, Mode 3)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
230h	V_RATE_INQ_1_4 (Format 1, Mode 4)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
234h	V_RATE_INQ_1_5 (Format 1, Mode 5)	FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps

Offset	Name	Field	Bit	Description
			[8-31]	Reserved
238h	V_RATE_INQ_1_6 (Format 1, Mode 6)	FrameRate_0	[0]	Reserved
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	240fps
			[8-31]	Reserved
23Ch	V_RATE_INQ_1_7 (Format 1, Mode 7)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
240h	V_RATE_INQ_2_0 (Format 2, Mode 0)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
244h	V_RATE_INQ_2_1 (Format 2, Mode 1)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps

Offset	Name	Field	Bit	Description
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
248h	V_RATE_INQ_2_2 (Format 2, Mode 2)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
24Ch	V_RATE_INQ_2_3 (Format 2, Mode 3)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
250h	V_RATE_INQ_2_4 (Format 2, Mode 4)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	Reserved
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
254h	V_RATE_INQ_2_5 (Format 2, Mode 5)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps

Offset	Name	Field	Bit	Description
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	120fps
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
258h	V_RATE_INQ_2_6 (Format 2, Mode 6)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
25Ch	V_RATE_INQ_2_7 (Format 2, Mode 7)	FrameRate_0	[0]	1.875fps
		FrameRate_1	[1]	3.75fps
		FrameRate_2	[2]	7.5fps
		FrameRate_3	[3]	15fps
		FrameRate_4	[4]	30fps
		FrameRate_5	[5]	60fps
		FrameRate_6	[6]	Reserved
		FrameRate_7	[7]	Reserved
			[8-31]	Reserved
260h : 2BFh	Reserved			
2E0h	V_CSR_INQ_7_0	Mode_0	[0-31]	CSR quadlet offset for Format_7 Mode_0
2E4h	V_CSR_INQ_7_1	Mode_1	[0-31]	CSR quadlet offset for Format_7 Mode_1
2E8h	V_CSR_INQ_7_2	Mode_2	[0-31]	CSR quadlet offset for Format_7 Mode_2
2ECh	V_CSR_INQ_7_3	Mode_3	[0-31]	CSR quadlet offset for Format_7 Mode_3

Offset	Name	Field	Bit	Description
2F0h	V_CSR_INQ_7_4	Mode_4	[0-31]	CSR quadlet offset for Format_7 Mode_4
2F4h	V_CSR_INQ_7_5	Mode_5	[0-31]	CSR quadlet offset for Format_7 Mode_5
2F8h	V_CSR_INQ_7_6	Mode_6	[0-31]	CSR quadlet offset for Format_7 Mode_6
2FCh	V_CSR_INQ_7_7	Mode_7	[0-31]	CSR quadlet offset for Format_7 Mode_7
300h	V_CSR_INQ_7_8	Mode_8	[0-31]	CSR quadlet offset for Format_7 Mode_8
304h	V_CSR_INQ_7_9	Mode_9	[0-31]	CSR quadlet offset for Format_7 Mode_9
308h	V_CSR_INQ_7_10	Mode_10	[0-31]	CSR quadlet offset for Format_7 Mode_10
30Ch	V_CSR_INQ_7_11	Mode_11	[0-31]	CSR quadlet offset for Format_7 Mode_11
310h	V_CSR_INQ_7_12	Mode_12	[0-31]	CSR quadlet offset for Format_7 Mode_12
314h	V_CSR_INQ_7_13	Mode_13	[0-31]	CSR quadlet offset for Format_7 Mode_13
318h	V_CSR_INQ_7_14	Mode_14	[0-31]	CSR quadlet offset for Format_7 Mode_14
31Ch	V_CSR_INQ_7_15	Mode_15	[0-31]	CSR quadlet offset for Format_7 Mode_15
320h	V_CSR_INQ_7_16	Mode_16	[0-31]	CSR quadlet offset for Format_7 Mode_16
324h	V_CSR_INQ_7_17	Mode_17	[0-31]	CSR quadlet offset for Format_7 Mode_17
328h	V_CSR_INQ_7_18	Mode_18	[0-31]	CSR quadlet offset for Format_7 Mode_18
32Ch	V_CSR_INQ_7_19	Mode_19	[0-31]	CSR quadlet offset for Format_7 Mode_19
330h	V_CSR_INQ_7_20	Mode_20	[0-31]	CSR quadlet offset for Format_7 Mode_20
334h	V_CSR_INQ_7_21	Mode_21	[0-31]	CSR quadlet offset for Format_7 Mode_21
338h	V_CSR_INQ_7_22	Mode_22	[0-31]	CSR quadlet offset for Format_7 Mode_22
33Ch	V_CSR_INQ_7_23	Mode_23	[0-31]	CSR quadlet offset for Format_7 Mode_23

Offset	Name	Field	Bit	Description
340h	V_CSR_INQ_7_24	Mode_24	[0-31]	CSR quadlet offset for Format_7 Mode_24
344h	V_CSR_INQ_7_25	Mode_25	[0-31]	CSR quadlet offset for Format_7 Mode_25
348h	V_CSR_INQ_7_26	Mode_26	[0-31]	CSR quadlet offset for Format_7 Mode_26
34Ch	V_CSR_INQ_7_27	Mode_27	[0-31]	CSR quadlet offset for Format_7 Mode_27
350h	V_CSR_INQ_7_28	Mode_28	[0-31]	CSR quadlet offset for Format_7 Mode_28
354h	V_CSR_INQ_7_29	Mode_29	[0-31]	CSR quadlet offset for Format_7 Mode_29
358h	V_CSR_INQ_7_30	Mode_30	[0-31]	CSR quadlet offset for Format_7 Mode_30
35Ch	V_CSR_INQ_7_31	Mode_31	[0-31]	CSR quadlet offset for Format_7 Mode_31

Inquiry Registers for Basic Functions

The following registers show which IIDC-compliant 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/Re-transmit)
		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 no 0 = Factory setting memory 1 = Memory Ch 1 2 = Memory Ch 2 : N= Memory Ch N If 0000, user memory is not available.

Inquiry Registers for Feature Presence

The following registers show the presence of the IIDC-compliant 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_Expo- sure	[1]	Auto Exposure Control
		Sharpness	[2]	Sharpness Control
		White_Bal- ance	[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_Shad- ing	[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_Func- tion_Inq	-	[0]	Reserved

Offset	Name	Field	Bit	Description
		PIO	[1]	Parallel input/output control
		SIO	[2]	Serial Input/output control
		Strobe_Out-put	[3]	Strobe signal output
		Lookup_Table	[4]	Lookup table control
		-	[5-31]	Reserved
410h-47Fh	Reserved			
480h	Advanced_Feature_Inq	Advanced_Feature_Quadlet_Off-set	[0-31]	Quadlet offset of the advanced feature CSR's (see the Advanced Registers section) from the base address of initial register space. (Vendor unique)
484h	PIO_Con-trol_CSR_Inq	PIO_Con-trol_Qua-dlet_Offset	[0-31]	Quadlet offset of the PIO control CSR's (see the Par-allel Input/Output (PIO) section) from the base address of initial register space.
488h	SIO_Con-trol_CSR_Inq	SIO_Con-trol_Qua-dlet_Offset	[0-31]	Quadlet offset of the SIO control CSR's (see the Serial Port Input/Output (SIO) section) from the base address of initial register space.
48Ch	Strobe_Output_CSR_Inq	Strobe_Out-put_Qua-dlet_Offset	[0-31]	Quadlet offset of the strobe output signal CSR's (see the Strobe Signal Output section) from the base address of initial register space.
490h	Lookup_Table_CSR_Inq	Lookup_Table_Qua-dlet_Offset	[0-31]	Quadlet offset of the Lookup Table CSRs from the baes address of initial register space.

Inquiry Registers for Feature Elements

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

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

Format:

Offset	Name	Field	Bit	Description
500h	BRIGHTNESS_INQ	Presence_Inq	[0]	Presence of this feature
		Abs_Con-trol_Inq	[1]	Absolute value control
			[2]	Reserved
		One_Push_Inq	[3]	One push auto mode (controlled auto-matically by camera only once)

Offset	Name	Field	Bit	Description
		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 by camera)
		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_EXPO-SURE_INQ	Same format as the BRIGHTNESS_INQ register		
508h	SHARPNESS_INQ	Same format as the BRIGHTNESS_INQ register		
50Ch	WHITE_BALANCE_ INQ	Same format as the BRIGHTNESS_INQ register		
510h	HUE_INQ	Same format as the BRIGHTNESS_INQ register		
514h	SATURATION_INQ	Same format as the BRIGHTNESS_INQ register		
518h	GAMMA_INQ	Same format as the BRIGHTNESS_INQ register		
51Ch	SHUTTER_INQ	Same format as the BRIGHTNESS_INQ register		
520h	GAIN_INQ	Same format as the BRIGHTNESS_INQ register		
524h	IRIS_INQ	Same format as the BRIGHTNESS_INQ register		
528h	FOCUS_INQ	Same format as the BRIGHTNESS_INQ register		
52Ch	TEMPERATURE_ INQ	Same format as the BRIGHTNESS_INQ register		
530h	TRIGGER_INQ	Presence_ Inq	[0]	Presence of this feature
		Abs_Con-trol_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

Offset	Name	Field	Bit	Description
		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 ID=4-6
		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_Con- trol_Inq	[1]	Absolute value control
			[2]	Reserved
		One_ Push_Inq	[3]	One push auto mode (controlled auto- matically by camera only once)
		ReadOut_ Inq	[4]	Ability to read the value of this feature
		On_Off_Inq	[5]	Ability to switch feature ON and OFF

Offset	Name	Field	Bit	Description
			[6-7]	Reserved
		Min_Value	[8-19]	Minimum value for this feature control
		Max_Value	[20-31]	Maximum value for this feature control
538h	WHITE_SHD_INQ	Same format as the BRIGHTNESS_INQ register		
53Ch	FRAME_RATE_INQ	Same format as the BRIGHTNESS_INQ register		
540h : 57Ch	Reserved for other FEATURE_HI_INQ			
580h	ZOOM_INQ	Presence_ Inq	[0]	Presence of this feature
		Abs_Con-trol_Inq	[1]	Absolute value control
			[2]	Reserved
		One_ Push_Inq	[3]	One push auto mode (controlled auto-matically by camera 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 by camera)
		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
584h	PAN_INQ	Same format as the ZOOM_INQ register		
588h	TILT_INQ	Same format as the ZOOM_INQ register		
58Ch	OPTICAL_FILTER_ INQ	Same format as the ZOOM_INQ register		

Appendix C: Video Mode Control and Status Registers for Format_7

These registers provide Format_7, Mode_x (Partial Image Size Format) information.

Inquiry Registers for Format_7 CSR Offset Addresses

The following set of registers indicates the locations of the Format_7 Mode_x CSR registers. These offsets are relative to the base offset 0xFFFF F0F0 0000.

Offset	Name	Field	Bit	Description
2E0h	V_CSR_INQ_7_0	Mode_0	[0-31]	CSR quadlet offset for Format_7 Mode_0
2E4h	V_CSR_INQ_7_1	Mode_1	[0-31]	CSR quadlet offset for Format_7 Mode_1
2E8h	V_CSR_INQ_7_2	Mode_2	[0-31]	CSR quadlet offset for Format_7 Mode_2
2ECh	V_CSR_INQ_7_3	Mode_3	[0-31]	CSR quadlet offset for Format_7 Mode_3
2F0h	V_CSR_INQ_7_4	Mode_4	[0-31]	CSR quadlet offset for Format_7 Mode_4
2F4h	V_CSR_INQ_7_5	Mode_5	[0-31]	CSR quadlet offset for Format_7 Mode_5
2F8h	V_CSR_INQ_7_6	Mode_6	[0-31]	CSR quadlet offset for Format_7 Mode_6
2FCh	V_CSR_INQ_7_7	Mode_7	[0-31]	CSR quadlet offset for Format_7 Mode_7
300h	V_CSR_INQ_7_8	Mode_8	[0-31]	CSR quadlet offset for Format_7 Mode_8

Table C.1: Format_7 Inquiry Register Offset Addresses

Current Format_7 Register Offsets

The actual offsets that are derived using the quadlet offset information in *Inquiry Registers for Format_7 CSR Offset Addresses* (page 144) are as follows:

The following table of Format_7 offsets is current as of the revision date. These offsets are subject to change without notice.

Offset Range	Description
2000h - 207Ch	Format_7 Mode_0 register offsets
2080h - 20FCh	Format_7 Mode_1 register offsets
2100h - 217Ch	Format_7 Mode_2 register offsets
2180h - 21FCh	Format_7 Mode_3 register offsets
2200h - 227Ch	Format_7 Mode_4 register offsets
2280h - 22FCh	Format_7 Mode_5 register offsets

Offset Range	Description
2300h - 237Ch	Format_7 Mode_6 register offsets
2380h - 23FCh	Format_7 Mode_7 register offsets
2400h - 247Ch	Format_7 Mode_8 register offsets

FORMAT_7_RESIZE_INQ: 1AC8h

This register reports all internal camera processes being used to generate images in the current Format 7 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: N/A 1: Available
	[1-7]	Reserved
Num_Cols	[8-11]	Number of columns being binned / subsampled, minus one e.g. if combining four columns together, this register will report a value of three.
Num_Rows	[12-15]	Number of rows binned / subsampled, minus one e.g. if combining four columns together, this register will report a value of three.
	[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

MAX_IMAGE_SIZE_INQ: 000h

This register is an inquiry register for maximum image size.

Format:

Field	Bit	Description
Hmax	[0-15]	Maximum horizontal pixel number
Vmax	[16-31]	Maximum vertical pixel number

UNIT_SIZE_INQ (004h) and UNIT_POSITION_INQ (04Ch)

This register is an inquiry register for unit size.

Hmax = Hunit * n = Hposunit*n3 (n, n3 are integers)
Vmax = Vunit * m = Vposunit*m3 (m, m3 are integers)

If the read value of Hposunit is 0, Hposunit = Hunit for compatibility with IIDC Rev 1.20.

If the read value of Vposunit is 0, Vposunit = Vunit for compatibility with IIDC Rev 1.20.

Format (UNIT_SIZE_INQ: 004h):

Field	Bit	Description
Hunit	[0-15]	Horizontal unit pixel number
Vunit	[16-31]	Vertical unit pixel number

Format (UNIT_POSITION_INQ: 04Ch):

Field	Bit	Description
Hposunit	[0-15]	Horizontal unit pixel number for position If read value of Hposunit is 0, Hposunit = Hunit for compatibility.
Vposunit	[16-31]	Vertical unit number for position If read value of Vposunit is 0, Vposunit = Vunit for compatibility.

IMAGE_POSITION (008h) and IMAGE_SIZE (00Ch)

These registers determine an area of required data. All the data must be as follows:

Left = Hposunit * n1
Top = Vposunit * m1

Width = Hunit * n2
Height = Vunit * m2 (n1, n2, m1, m2 are integers)
Left + Width <= Hmax
Top + Height <= Vmax

Format (IMAGE_POSITION: 008h):

Field	Bit	Description
Left	[0-15]	Left position of requested image region (pixels)
Top	[16-31]	Top position of requested image region (pixels)

Format (IMAGE_SIZE: 00Ch):

Field	Bit	Description
Width	[0-15]	Width of requested image region (pixels)
Height	[16-31]	Height of requested image region (pixels)

COLOR_CODING_ID (010h) and COLOR_CODING_INQ (014h)

The COLOR_CODING_INQ register describes available 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 (COLOR_CODING_ID: 010h):

Field	Bit	Description
Coding_ID	[0-7]	Color coding ID from COLOR_CODING_INQ register
	[8-31]	Reserved (all zero)

Format (COLOR_CODING_INQ: 014h):

Field	Bit	Description	ID
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 (all zero)	11-31

PIXEL_NUMBER_INQ (034h), TOTAL_BYTES_HI_INQ (038h), and TOTAL_BYTES_LO_INQ (03Ch)

The PIXEL_NUMBER_INQ register includes the total number of pixels in the required image area. The TOTAL_BYTE_INQ register includes the total number of bytes in the required image area.

If the *Presence* bit in the VALUE_SETTING register(page 149) is zero, the values of these registers will be updated by writing the new value to the IMAGE_POSITION (page 146), IMAGE_SIZE (page 146) and COLOR_CODING_ID (page 147) registers.

If the *Presence* bit in the VALUE_SETTING register is one, the values of these registers 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 registers are valid.

Format (PIXEL_NUMBER_INQ: 034h):

Field	Bit	Description
PixelPerFrame	[0-31]	Pixel number per frame

Format (TOTAL_BYTES_HI_INQ: 038h):

Field	Bit	Description
BytesPerFrameHi	[0-31]	Higher quadlet of total bytes of image data per frame

Format (TOTAL_BYTES_LO_INQ: 03Ch):

Field	Bit	Description
BytesPerFrameLo	[0-31]	Lower quadlet of total bytes of image data per frame

PACKET_PARA_INQ (040h) and BYTE_PER_PACKET (044h)

MaxBytePerPacket describes the maximum packet size for one isochronous packet.
UnitBytePerPacket is the unit for isochronous packet size.
RecBytePerPacket describes the recommended packet size for one isochronous packet. If *RecBytePerPacket* is zero, you must ignore this field.

If the *Presence* bit in the VALUE_SETTING register (page 149) is zero, values of these fields will be updated by writing the new value to the IMAGE_POSITION (page 146), IMAGE_SIZE (page 146) and COLOR_CODING_ID (page 147) registers with the value of the ISO_Speed register (60Ch [6..7]).

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.

The *BytePerPacket* value determines the real packet size and transmission speed for one frame image. The *BytePerPacket* value must keep the following condition.

BytePerPacket = UnitBytePerPacket * n (n is an integer)

BytePerPacket <= MaxBytePerPacket

Format (PACKET_PARA_INQ: 040h):

Field	Bit	Description
UnitBytePerPacket	[0-15]	Minimum bytes per packet
MaxBytePerPacket	[16-31]	Maximum bytes per packet

Format (BYTE_PER_PACKET: 044h):

Field	Bit	Description
BytePerPacket	[0-15]	Packet size
RecBytePerPacket	[16-31]	Recommended bytes per packet. If this value is zero, must ignore this field.

PACKET_PER_FRAME_INQ: 048h

If *BytePerPacket * n != BytePerFrame* (n is an integer), you must use padding. The PacketPerFrame value is the number of packets per one frame. This register will be updated after BytePerPacket is written.

The total number of bytes of transmission data per one frame = *BytePerPacket * PacketPerFrame*.

The number of bytes of padding = *BytePerPacket * PacketPerFrame – BytePerFrame*. The receiver must ignore the above padding data in the last packet of each frame.

Format:

Field	Bit	Description
PacketPerFrame	[0-31]	Number of packets per frame

FRAME_INTERVAL_INQ: 050h

This register describes the frame interval based on the current camera conditions, including exposure time. The reciprocal value of this (1 / FrameInterval) is the frame rate of the camera. If the value of this register is zero, the camera can't report this value and it should be ignored. FrameInterval is in seconds and reported in IEEE1394/REAL*4 floating- point format (see *Determining Absolute Value Register Values*(page 154)

Format:

Field	Bit	Description
FrameInterval	[0-31]	Current frame interval (seconds) (IEEE/REAL*4 floating-point value) If read value of FrameInterval is zero, ignore this field.

VALUE_SETTING: 07Ch

The purpose of the *Setting_1* bit is for updating the TOTAL_BYTES_HI_INQ (page 148), TOTAL_BYTES_LO_INQ (page 148) PACKET_PARA_INQ (page 148) and BYTE_PER_PACKET (page 148) registers. If one of the values in the IMAGE_POSITION (page 146), IMAGE_SIZE(page 146) COLOR_CODING_ID (page 147) and ISO_SPEED registers is changed, the *Setting_1* bit must be set to 1. The *ErrorFlag_1* field will be updated when the *Setting_1* bit returns to 0. If the *ErrorFlag_1* field is zero, the values of the TOTAL_BYTES_HI_INQ, TOTAL_BYTES_LO_INQ, PACKET_PARA_INQ and BYTE_PER_PACKET registers are valid.

After the *BytePerPacket* value is written, the *ErrorFlag_2* field will be updated. If the *ErrorFlag_2* field is zero, isochronous transmission can be started without any problem.

Format:

Field	Bit	Description
Presence	[0]	If this bit is one, 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 when Setting_1 bit returns to "0" from "1".
ErrorFlag_2	[9]	BytePerPacket value is not acceptable. 0: no error, 1: error
	[10-31]	Reserved

Appendix D: Absolute Value Registers

Inquiry Registers for Absolute Value CSR Offset Addresses

The following set of registers indicates the locations of the absolute value CSR registers. These offsets are relative to the base offset 0xFFFF F0F0 0000.

Offset	Name	Bit	Description
700h	ABS_CSR_HI_INQ_0	[0..31]	Quadlet offset for the absolute value CSR for Brightness.
704h	ABS_CSR_HI_INQ_1	[0..31]	Quadlet offset for the absolute value CSR for Auto Exposure.
708h	ABS_CSR_HI_INQ_2	[0..31]	Quadlet offset for the absolute value CSR for Sharpness.
70Ch	ABS_CSR_HI_INQ_3	[0..31]	Quadlet offset for the absolute value CSR for White Balance.
710h	ABS_CSR_HI_INQ_4	[0..31]	Quadlet offset for the absolute value CSR for Hue.
714h	ABS_CSR_HI_INQ_5	[0..31]	Quadlet offset for the absolute value CSR for Saturation.
718h	ABS_CSR_HI_INQ_6	[0..31]	Quadlet offset for the absolute value CSR for Gamma.
71Ch	ABS_CSR_HI_INQ_7	[0..31]	Quadlet offset for the absolute value CSR for Shutter.
720h	ABS_CSR_HI_INQ_8	[0..31]	Quadlet offset for the absolute value CSR for Gain.
724h	ABS_CSR_HI_INQ_9	[0..31]	Quadlet offset for the absolute value CSR for Iris.
728h	ABS_CSR_HI_INQ_10	[0..31]	Quadlet offset for the absolute value CSR for Focus.
72Ch	ABS_CSR_HI_INQ_11	[0..31]	Quadlet offset for the absolute value CSR for Temperature.
730h	ABS_CSR_HI_INQ_12	[0..31]	Quadlet offset for the absolute value CSR for Trigger.
734h	ABS_CSR_HI_INQ_13	[0..31]	Quadlet offset for the absolute value CSR for Trigger Delay.
73Ch	ABS_CSR_HI_INQ_15	[0..31]	Quadlet offset for the absolute value CSR for Frame Rate.
740h - 77Fh	Reserved		

Offset	Name	Bit	Description
7C4h	ABS_CSR_LO_INQ_1	[0..31]	Quadlet offset for the absolute value CSR for Pan.
7C8h	ABS_CSR_LO_INQ_2	[0..31]	Quadlet offset for the absolute value CSR for Tilt.

Current Absolute Value Register Offsets

The absolute value offsets that would be derived using the quadlet offset information in the Inquiry Registers for Absolute Value CSR Offset Addresses section would be as follows:

These offsets are current as of the revision date. They are subject to change without notice.

Offset	Name	Field	Bit	Description
900h	ABS_VAL_AUTO_EXPOSURE	Min_Value	[0-31]	Min auto exposure value.
904h		Max_Value	[0-31]	Max auto exposure value.
908h		Value	[0-31]	Current auto exposure value.
910h	ABS_VAL_SHUTTER	Min_Value	[0-31]	Min shutter value seconds
914h		Max_Value	[0-31]	Max shutter value seconds
918h		Value	[0-31]	Current shutter value seconds
920h	ABS_VAL_GAIN	Min_Value	[0-31]	Min gain value dB
924h		Max_Value	[0-31]	Max gain value dB
928h		Value	[0-31]	Current gain value dB
930h	ABS_VAL_BRIGHTNESS	Min_Value	[0-31]	Min brightness value %
934h		Max_Value	[0-31]	Max brightness value %
938h		Value	[0-31]	Current brightness value %
940h	ABS_VAL_GAMMA	Min_Value	[0-31]	Min gamma value
944h		Max_Value	[0-31]	Max gamma value
948h		Value	[0-31]	Current gamma value
950h	ABS_VAL_TRIGGER_DELAY	Min_Value	[0-31]	Min delay value seconds
954h		Max_Value	[0-31]	Max delay value seconds
958h		Value	[0-31]	Current delay value seconds
960h	ABS_VAL_FRAME_RATE	Min_Value	[0-31]	Min frame rate FPS
964h		Max_Value	[0-31]	Max frame rate FPS
968h		Value	[0-31]	Current frame rate FPS
970h	ABS_VAL_HUE	Min_Value	[0-31]	Min hue deg
974h		Max_Value	[0-31]	Max hue deg

Offset	Name	Field	Bit	Description
978h		Value	[0-31]	Current hue deg
980h	ABS_VAL_SATURATION	Min_Value	[0-31]	Min saturation %
984h		Max_Value	[0-31]	Max saturation %
988h		Value	[0-31]	Current saturation %

Absolute Value Register Format

The IIDC Specification defines the Absolute Value CSRs. Each set of absolute value CSRs consists of three registers (quadlets): a minimum value, a maximum value (both read only) and the current value. The IIDC specification defines the offsets of the Absolute Value CSRs for each of the camera features, as described in *Current Absolute Value Register Offsets* on previous page

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

0-7	8-15	16-23	24-31
Floating-point value with IEEE/REAL*4 format			

Sign(S)	Exponent(exp)	Mantissa(m)
1bit	8bit	23bit

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 element name	Function	Unit	Unit Descrip-tion	Reference point	Value Type
Brightness	Black level offset	%		----	Absolute
Auto Exposure	Auto Exposure	EV	exposure value	0	Relative
White_Balance	White Balance	K	kelvin	----	Absolute
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
Focus	Focus	m	meter	----	Absolute

Feature element name	Function	Unit	Unit Description	Reference point	Value Type
Trigger	External Trigger	times		----	Absolute
Trigger_Delay	Trigger Delay	S	seconds	----	Absolute
Frame_Rate	Frame rate	fps	frames per second	----	Absolute

Determining Absolute Value Register Values

The Absolute Value CSRs store 32-bit floating-point values with IEEE/REAL*4 format, as described in Absolute Value Register Format on previous page 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 quadlet offset for the absolute value CSR for shutter.

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

2. The 32-bit ulValue is a quadlet 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.*

Setting Absolute Value Register Values

The user must write a 1 to bit [1] of the associated feature CSR to change the Value field of this register from being read-only. For example, to enable absolute value control of shutter, bit [1] of SHUTTER register 0x81C must be set to 1. In the FlyCapture API, this can also be done by setting the `absControl` member of the of the desired property structure to true.

Appendix E: 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 1394 host adapter in the PC). Every video format, mode and frame rate has a different video data format.

Isochronous Packet Format for Format_0, Format_1 and Format_2

The following table shows the format of the first quadlet (a quadlet being four bytes) in the data field of an isochronous data block.

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

Table E.1: Isochronous Data Packet Formatfor Format_0, Format_1 and Format_2

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.

Isochronous Bandwidth Requirements

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 quadlets 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)	8)8H 2560p	4)4H 1280p	2H 640p	1H 320p	1/2H 160p	1/4H 80p	1/8H 40p	1/16H 20p

Mode	Video Format	240fps	120fps	60fps	30fps	15fps	7.5fps	3.75fps	1.875fps
	16bit/pixel	1280q	640q	320q	160q	80q	40q	20q	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		
2	800x600 Y (Mono) 8bit/pixel	16)20H 16000p 4000q	8)10H 8000p 2000q	4)5H 4000p 1000q	2)5/2H 2000p 500q	5/4H 1000p 250q	5/8H 500p 125q		
3	1024x768 YUV(4:2:2) 16bit/pixel		32)12H 12288p 6144q	16)6H 6144p 3072q	8)3H 3072p 1536q	4)3/2H 1536p 768q	2)3/4H 768p 384q	3/8H 384p 192q	3/16H 192p 96q
4	1024x768 RGB 24bit/pixel			32)6H 6144p 4608q	16)3H 3072p 2304q	8)3/2H 1536p 1152q	4)3/4H 768p 576q	2)3/8H 384p 288q	3/16 192p 144q
5	1024x768 Y (Mono) 8bit/pixel	32)24H 24576p 6144q	16)12H 12288p 3072q	8)6H 6144p 1536q	4)3H 3072p 768q	2)3/2H 1536p 384q	3/4H 768p 192q	3/8H 384p 96q	3/16H 192p 48q
6	800x600 Y (Mono16) 16bit/pixel	32)20H 16000p 8000q	16)10H 8000p 4000q	8)5H) 4000p 2000q	4)5/2H 2000p 1000q	2)5/4H 1000p 500q	5/8H 500p 250q	5/16H 250p 125q	
7	1024x768 Y (Mono16) 16bit/pixel		32)12H 12288p 6144q	16)6H 6144p 3072q	8)3H 3072p 1536q	4)3/2H 1536p 768q	2)3/4H 768p 384q	3/8H 384p 192q	3/16H 192p 96q

Format_2

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

[--H – Lines/Packet]
[--p – Pixels/Packet]
[--q – Quadlets/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

Isochronous Packet Format for Format_7

The following table shows the format of the first quadlet (a quadlet being four bytes) in the data field of an isochronous data block.

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

Table E.2: Isochronous Data Packet Format for Format_7

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.

Appendix F: Glossary

Term	Definition
1394a	An Institute of Electrical and Electronics Engineers (IEEE) interface standard capable of transferring data at a rate of 400Mbit per second.
1394b	An IEEE interface standard capable of transferring data at a rate of 800Mbit per second.
Absolute Values	Real-world values, such as milliseconds (ms), decibels (dB) or percent (%). Using the absolute values is easier and more efficient than applying complex conversion formulas to integer values.
Analog-to-Digital Converter	Often abbreviated as ADC orA/D converted, it is a device that converts a voltage to a digital number.
API	Application Programming Interface. Essentially a library of software functions.
Asynchronous Transmission	The transfer of image data from the camera to the PC that is regulated by an external signal, such as a trigger. Asynchronous transfers do not guarantee when data will be transferred. However, they do guarantee that data will arrive as sent.Asynchronous transfers may be used when data integrity is a higher priority than speed. An example might be an image data transfer to a printer, where speed is less critical than getting the image pixels correct. Asynchronous transfers are initiated from a single node, designated the ‘requestor’, to or from the address space of another node, designated the ‘responder’. Asynchronous requests are packet- based. The requestor node generates a request packet that the 1394 bus sends to the responder node. The responder node is responsible for handling the request packet and creating a response packet that is sent back to the requestor node to complete a single transfer. There are three types of 1394 asynchronous transfers: Read, Write and Lock.
BPP	Bytes per packet. An image is broken into multiple packets of data, which are then streamed isochronously to the host system. Each packet is made up of multiple bytes of data.
Brightness (%)	This is essentially the level of black in an image. A high brightness will result in a low amount of black in the image. In the absence of noise, the minimum pixel value in an image acquired with a brightness setting of 1% should be 1% of the A/D converter’s minimum value.
Config ROM	Configuration read-only memory. A section of memory dedicated to describing low-level device characteristics such as Model and Vendor ID, IEEE-1394 version compliance, base address quadlet offsets, etc.
Color Processing	Also known as ‘interpolation,’ an algorithm for converting raw Bayer-tiled image data into full color images. Depending on camera model, this process takes place either on-camera or on the PC. For more information, refer to Knowledge Base Article 33.
Dynamic Range	The difference between the maximum and minimum amounts of light that a sensor can measure. This is bounded on the upper end by the maximum charge that any pixel can contain (sensor full well depth) and at the lower end by the small charge that every sensor spontaneously generates (read noise).
Exposure (EV)	This is the average intensity of the image. It will use other available (non- manually adjustable) controls to adjust the image.

Term	Definition
Firmware	Programming that is inserted into programmable read-only memory, thus becoming a permanent part of a computing device. Firmware is created and tested like software and can be loaded onto the camera.
Format_7	Encompasses partial or custom image video formats and modes, such as region of interest of pixel binned modes. Format_7 modes and frame rates are defined by the camera manufacturer, as opposed to the IIDC specification.
FPS	Frames Per Second.
Frame Rate	Often defined in terms of number of frames per second (FPS) or frequency (Hz). This is the speed at which the camera is streaming images to the host system. It basically defines the interval between consecutive image transfers.
Gain (dB)	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 and an increase in noise.
Gamma	Gamma defines the function between incoming light level and output picture level. Gamma can also be useful in emphasizing details in the darkest and/or brightest regions of the image.
GPIO	General Purpose Input/Output.
Grabbing Images	A commonly-used phrase to refer to the process of enabling isochronous transfers on a camera, which allows image data to be streamed from the camera to the host system.
Hz	Hertz. A unit of frequency; one Hertz has a periodic interval of one second. Often used interchangeably with FPS as a measure of frame rate.
Isochronous Transmission	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 1394 bus guarantee timely delivery of data. Specifically, isochronous transfers are scheduled by the bus so that they occur once every 125µs. Each 125µs timeslot on the bus is called a frame. Isochronous transfers, unlike asynchronous transfers, do not guarantee the integrity of data through a transfer. No response packet is sent for an isochronous transfer. Isochronous transfers are useful for situations that require a constant data rate but not necessarily data integrity. Examples include video or audio data transfers. Isochronous transfers on the 1394 bus do not target a specific node. Isochronous transfers are broadcast transfers which use channel numbers to determine destination.
Lookup Table	A matrix of gamma functions for each color value of the current pixel encoding format.
Node	An addressable device attached to a bus. Although multiple nodes may be present within the same physical enclosure (module), each has its own bus interface and address space and may be reset independently of the others.
Node ID	A 16-bit number that uniquely differentiates a node from all other nodes within a group of interconnected buses. Although the structure of the node ID is bus-dependent, it usually consists of a bus ID portion and a local ID portion. The most significant bits of the node ID are the same for all nodes on the same bus; this is the bus ID. The least-significant bits of the node ID are unique for each node on the same bus; this is called the local ID. The local ID may be assigned as a consequence of bus initialization.
One Push	For use when a control is in manual adjust mode, One Push sets a parameter to an auto-adjusted value, then returns the control to manual adjust mode.
PHY	Physical layer. Each 1394 PHY provides the interface to the 1394 bus and performs key functions in the communications process, such as bus configuration, speed signaling and detecting transfer speed, 1394 bus control arbitration, and others.

Term	Definition
Pan	A mechanism to horizontally move the current portion of the sensor that is being imaged. In stereo and spherical cameras, Pan controls which individual sensors transmit images.
Pixel Clock	The rate at which the sensor outputs voltage signals in each pixel from the optical input.
Pixel Format	The encoding scheme by which color or greyscale images are produced from raw image data.
Quadlet	A 4 byte (32-bit) value.
Quadlet Offset	The number of quadlets separating a base address and the desired CSR address. For example, if the base address is 0xFFFFF0F00000 and the value of the quadlet offset is 0x100, then the actual address offset is 0x400 and the actual adress 0xFFFFF0F00400.
Register	A term used to describe quadlet-aligned addresses that may be read or written by bus transactions.
Saturation	This is how far a color is from a gray image of the same intensity. For example, red is highly saturated, whereas a pale pink is not.
SDK	Software Development Kit
Sharpness	This works by filtering the image to reduce blurred edges.
Shutter	A mechanism to control the length of time the sensor is exposed to light from the image field for each frame. In milliseconds (ms), it is the amount of time that the shutter stays open, also known as the exposure or integration time. The shutter time defines the start and end point of when light falls on the imaging sensor. At the end of the exposure period, all charges are simultaneously transferred to light-shielded areas of the sensor. The charges are then shifted out of the light shielded areas of the sensor and read out.
Signal-to-Noise Ratio (dB)	The difference between the ideal signal that you expect and the real-world signal that you actually see is usually called noise. The relationship between signal and noise is called the signal-to- nose ratio (SNR). SNR is calculated using the general methodology outlined in Knowledge Base Article 142.
SXGA	1280x1024 pixel resolution
Tilt	A mechanism to vertically move the current portion of the sensor that is being imaged.
Trigger	A signal to which the acquisition of images by the camera is synchronized. Triggers can be from an outside electrical source (external) or software-generated (internal).
UXGA	1600x1200 pixel resolution
VGA	640x480 pixel resolution
White Balance	A method to enable white areas of an image to appear correctly by modifying the gain of red and blue channels relative to the green channel. White balance can be used to accommodate differing lighting conditions.
XVGA	1024x768 pixel resolution

Appendix G: Contacting Point Grey Research

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

Email:	For all general questions about Point Grey Research please contact us at info@ptgrey.com. For technical support (existing customers only) contact us at www.ptgrey.com/support/contact/.	
Knowledge Base:	Find answers to commonly asked questions in our knowledge base at www.ptgrey.com/support/kb/.	
Downloads:	Users can download the latest manuals and software from www.ptgrey.com/support/downloads/ .	
Main Office:	**Mailing Address:** Point Grey Research, Inc. 12051 Riverside Way Richmond, BC, Canada V6W 1K7	**Tel**: +1 (604) 242-9937 **Toll-free** (North America only): +1 (866) 765-0827 **Fax**: +1 (604) 242-9938 sales@ptgrey.com
Distributors		
USA	**Tel**: +1 (866) 765-0827 na-sales@ptgrey.com	
Europe & Israel	**Mailing Address**: Point Grey Research GmbH Schwieberdinger Strasse 60 71636 Ludwigsburg Germany	**Tel**: +49 7141 488817-0 **Fax**: +49 7141 488817-99 eu-sales@ptgrey.com
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 and Thailand	Voltrium Systems Pte Ltd. (www.voltrium.com.sg)	
Taiwan	Apo Star Co., Ltd. (www.apostar.com.tw)	

Appendix H: Revision History

Revision	Date	Notes
1.0	February 22, 2011	• Initial version

Index