MAKING DIGITAL IMAGING SIMPLE

PL-A741 Machine Vision Camera

System Guide

Version 4.0

PixeLINK
3030 Conroy Road • Ottawa, Ontario • K1G 6C2 • Canada • Tel: (613) 247-1211 • Fax: (613) 247-2001
Email: **info@pixelink.com** • Web Site: **www.pixelink.com**

PixeLINK™ PL-A741 Machine Vision Camera System Guide

Copyright Notice

Version 4.0

This document contains proprietary and confidential information of PixeLINK. The contents of this document may not be copied nor duplicated in any form, in whole or in part, without prior written consent from PixeLINK.

By purchasing this product, the Purchaser(s) and/or any subsequent legitimate owner(s) of the product, henceforth referred to as "the Purchaser," agree(s) to abide by the terms of this Agreement and read and recognize the following set of definitions appertaining to the intellectual-property items and trademark references as can be found throughout this System Guide.

PixeLINK provides the information and data included in this document for the Purchaser's benefit, but it is not possible for PixeLINK to entirely verify and test all of this information in all circumstances, particularly information relating to non-PixeLINK manufactured products. PixeLINK makes no warranties or representations relating to the quality, content, or adequacy of this information. Every effort has been made to ensure the accuracy of this Guide; however, PixeLINK assumes no responsibility for any errors or omissions in this document. PixeLINK shall not be held liable for any errors or for incidental or consequential damages in connection with the furnishing, performance, or use of this System Guide or the examples herein.

PixeLINK assumes no responsibility for any damage or loss resulting from the use of this System Guide, loss or claims by third parties which may arise through the use of this product, any damage or loss caused by deletion of data as a result of malfunction or repair, or any other damage related to the use of this product or associated documentation. The information in this document is subject to change without notice.

Definitions of Intellectual Property and Trademark Attributions

This Section is intended to ensure proper attribution and honoring of any and all trademarks and intellectual-property items in terms of attribution to their respective owners as mentioned in this System Guide. The reader is encouraged to consult this Section whenever uncertainty presents itself as to the terms, their meaning within the System Guide, and the trademarks and intellectual-property items they stand to identify, whether by themselves or in conjunction with other terms and items.

PixeLINK is either a trademark or a registered trademark of PixeLINK in Canada and/or other countries; *IEEE* is a registered trademark or service mark of the Institute of Electrical and Electronics Engineers, Incorporated in the United States and/or other countries; *FireWire* is a trademark of Apple Computer, Inc., registered in the U.S. and other countries; *Microsoft, DirectShow,* and *Windows* are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries; *TIFF* is a trademark or registered trademark of Adobe Systems Incorporated in the United States and/or other countries. All other products, brand names, company names are trademarks or registered trademarks of their respective owners.

Definition of Terms

This Section is intended to define certain terminology used in this System Guide, while ensuring proper attribution and honoring of any and all trademarks and intellectual-property items in terms of attribution to their respective owners as mentioned in this System Guide.

The Purchaser shall hereby recognize the following definitions set herein, as can be found throughout this System Guide: *Camera* shall henceforth refer to a PixeLINK Camera; *API* shall henceforth refer to the PixeLINK Camera Application Programming Interface; *Kit* shall henceforth refer to a PixeLINK Camera Kit; *FireWire* shall henceforth refer to the IEEE 1394a interface specification; *DirectShow* shall henceforth refer to the Microsoft DirectShow multimedia software; *Windows* shall henceforth refer to a Microsoft Windows operating system.

The above Sections set forth Terms and Conditions, compliance with which constitutes a mandatory prerequisite for owning and/or using the product for which the Guide was created. It is the Purchaser's responsibility to ensure that the information contained within the Sections is maintained as a part of the System Guide at all times—should the Purchaser discover that the page(s) containing the Sections is (are) missing, and/or was not provided with the System Guide, and/or become illegible, PixeLINK should be contacted as soon as possible and the Sections requested. PixeLINK shall not be held liable for any and all copyright violations that may ensue in relation to its products and/or the consequences of their intended and unintended usage.

October 2003
Part Number: 04327-01

Medical Use

This camera is not intended for use in medical applications.

Table of Contents

List of Figures

List of Tables

1

Introduction

1.1 The PixeLINK PL-A741 Machine Vision Camera

The PL-A741 is a high-performance, 1.3 megapixel monochrome C-mount camera designed specifically for machine vision applications. The camera is connected to the computer by a FireWire interface and is fully IIDC 1.3 compatible. In addition, the camera provides non-IIDC features that raise it above and beyond the standard, providing excellent performance for the price.

Camera Features:

- No Framegrabber Required
- FireWire Interface (Two Connectors)
- IIDC 1.3 (DCAM) Compatible … PLUS Extended Features
- 33 fps Uncompressed at 1k × 1k
- Extended Dynamic Range
- Global Shutter—Frame on Demand
- Knee Points (Multiple Slope Exposure)
- External Trigger
- Available in "Right Angle" configuration

Custom applications can be developed on Windows platforms in C/C++ or Visual Basic with the PixeLINK Camera Application Programming Interface (API), available in the PixeLINK Camera SDK and Developer's Kits (Version 4). The PixeLINK API offers more functionality and greater control than using the PL-A741 with IIDC features alone. Visit http://ww.pixelink.com/ or contact your PixeLINK vendor for details.

1.2 Topics Covered in this Guide

This Guide provides a reference for the PL-A741 hardware and connectors, and the control options available through IIDC 1.3.

Users should consult this Guide …

… before using the PL-A741 with an IIDC-compliant application

… when mounting the PL-A741 within a system or enclosure

… when planning to use an external trigger or GPO device (e.g., strobe, pulse) with the PL-A741

Sections include:

- A description of the PL-A741 hardware—Section 2
- An overview of IIDC compliance and features available with the PL-A741—Section 3
- A list of available triggering and GPO modes—Section 4
- IIDC features—Sections 5 to 9
- A description of the PL-A741's operating (exposure) modes and shutter types—Section 10

References in the PDF version of this Guide are hyperlinked for easy navigation and access.

1.3 System Requirements

Hardware Requirements

- Microprocessor:
  Recommended—Pentium® 4, Mac G4, or equivalent, 1.5 GHz
  Minimum—Pentium® III, Mac G4, or equivalent, 450 MHz
- Memory:
  Recommended—128 MB RAM
  Minimum—64 MB RAM
- Desktop resolution:
  Recommended—1280 × 1024
  Minimum—VGA (640 × 480)
- 25 MB of hard drive space
- A video card with 24-bit or 32-bit True Color graphics capability, at least 8 MB of video memory and the latest vendor driver installed

Operating System Requirements for PixeLINK Software:

Microsoft Windows

The PL-A741 camera can be controlled by PixeLINK applications created using API Version 4 running on:

- Windows 2000, with Service Pack 1 (SP1) installed, or
- Windows XP, with Service Pack 1 (SP1) installed

Service Packs are available for download from the Microsoft Web Site, http://www.microsoft.com/

Laptop Computers Only

Because of the specialized hardware configurations of laptop computers, they require additional installation considerations:

- **A built-in FireWire port, or a preinstalled CardBus card slot and a FireWire CardBus adapter card (400 Mbps)**
- **An external 12 V DC power supply**—Although certain laptop models may be able to provide power to the camera via the FireWire bus, it is more likely that the system will require an external power supply for the camera. (Note: A four-pin FireWire connector cannot provide power to the camera, nor can a CardBus adapter card provide sufficient power.)

  For more information about laptop accessory kits including power supplies and CardBus cards, visit http://ww.pixelink.com/ or contact your PixeLINK vendor.

1.4 Functionality with Third-Party and Custom Applications

PixeLINK cameras can be controlled by the following kinds of applications:

- **Applications compliant with the IIDC 1.3 specification**

The PL-A741 can be used "out of the box" with IIDC 1.3 ("DCAM" or "Digital Camera") compliant applications. The PL-A741 uses IIDC features extensively for controlling the camera.

For more information about IIDC 1.3 compliance with the PL-A741, refer to Section 3 (on page 26).

- **Applications created using the PixeLINK Camera API (Version 4)**

(For Windows-based systems only) The PixeLINK Camera Application Programming Interface (API) offers application software developers a means to adapt existing programs or develop new imaging applications for PixeLINK cameras. It allows rapid development of custom applications for camera operation by simplifying the most common tasks associated with configuring and controlling the cameras.

The PixeLINK Camera API Version 4.0 is a dynamic link library (DLL) that provides:

- A powerful, easy to use control interface
- A generic command set that can be used to control different camera models
- Fast and flexible access to streaming video
- The ability to save high quality still images and video clips

The PixeLINK "extensions" in the API control functions not covered by the IIDC 1.3 specification, offering greater functionality than in standalone IIDC usage of the camera.

API functions may be called from C/C++ or Visual Basic.

The PixeLINK Camera API software and reference documentation are included in PixeLINK Camera Developer's Kits and SDKs.

FREE Demonstration Application

The *PixeLINK Developers Application* demonstrates all major features of the PL-A741 and the PixeLINK Camera API. This application is available for download from the PixeLINK Web site (http://www.pixelink.com/) and in PixeLINK Camera SDK and Developer's Kits.

Purchasers of the SDK and Developer's Kits also receive sample code, simplifying integration of API functions into custom applications.

- **Third-party applications**

Third-party software vendors can maximize performance with the PL-A741 by using the PixeLINK Camera API to integrate camera controls directly into their applications. By integrating the controls directly, the application can access an extended range of options not available through IIDC. Check with the third party vendor to see if PixeLINK controls have been integrated into the software.

The PixeLINK Web site http://www.pixelink.com/ lists third-party applications recognized by PixeLINK as having been specially designed to work with PixeLINK cameras.

1.5 Compatibility with Other PixeLINK Products

The PL-A741 is compatible with other Version 4.0 PixeLINK Products.

1.6 Related PixeLINK Documentation

- **PL-A741 User's Manual**

  This manual describes the functionality of the PixeLINK Megapixel FireWire camera hardware and software, including the *PixeLINK Developers Application*.

  Users should consult the User's Manual

  … if this is the first time installing FireWire or USB hardware
  … before installing PixeLINK software
  … as a guide when using the *PixeLINK Developers Application*

  Sections include:

  - Installation of the camera hardware and PixeLINK software
  - Features and operation of the *PixeLINK Developers Application*

- **PixeLINK Camera API Reference, Version 4**

  This manual provides a reference for the PixeLINK Camera API and related software.

  Sections include:

  - Basics principles of using the PixeLINK Camera API

- A summary of the PixeLINK Camera API functions
- Descriptions of individual API functions
- Sample code

1.7 Camera Accessories

Accessories such as trigger kits, tripod mounts, lenses, laptop accessory kits and hands-free switches are available from PixeLINK. For a current list of accessories, visit the PixeLINK Web site at http://www.pixelink.com/ or contact your PixeLINK vendor.

2

Hardware Overview

2.1 PL-A741 Camera Hardware Features

Figure 2.1 PL-A741 FireWire Machine Vision Camera, Standard Configuration

Hardware Features:

- 1.3 Megapixel imager resolution (1280 × 1024)
- FireWire interface (two six-pin connectors)
- Machine Vision interface for external trigger and GPO (strobe)
- Lens mount for a standard C-mount lens (1" × 32 tpi) [lens not included]
- Front and bottom mounting holes for a tripod or other mounting fixture (4–40 screws)
- Two configurations: Standard and Right Angle

See Section 2.2 (on page 13) for hardware dimensions and locations of hardware features.

Accessories

Accessories such as trigger kits, tripod mounts, lenses, laptop accessory kits and hands-free switches are available from PixeLINK. For a current list of accessories, visit the PixeLINK Web site at http://www.pixelink.com/ or contact your PixeLINK vendor.

2.2 PL-A741 Camera Dimensions

Measurements are ± 0.005 inches (0.15 mm).

2.2.1 Standard Configuration

Height .... 1.38 in /35 mm
Width .... 1.97 in/50 mm
Length .... 3.94 in/100 mm
Weight .... 5.6 oz/160 g (without lens)

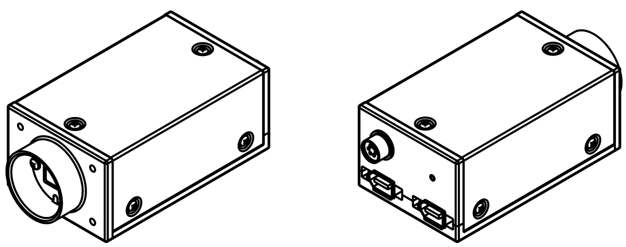


Figure 2.2 Standard Configuration

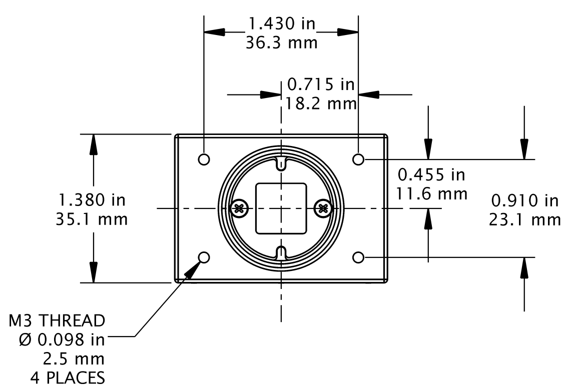


Figure 2.3 Standard Configuration, Front View

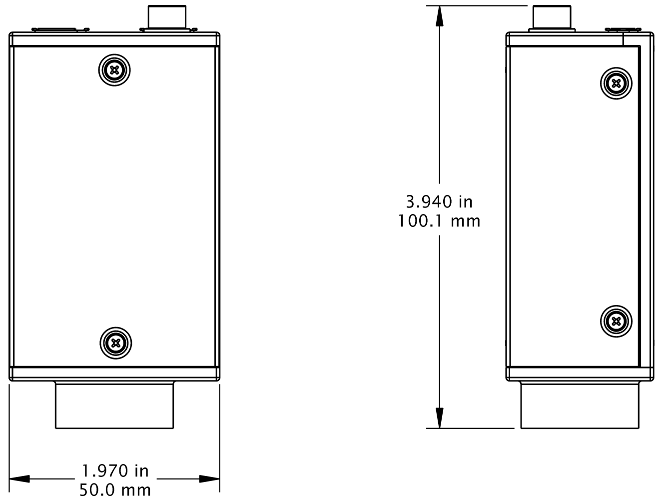


Figure 2.4 Standard Configuration, Top and Side Views

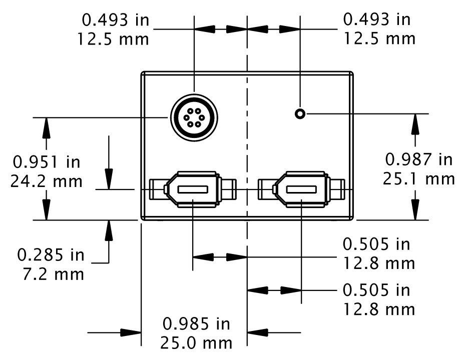


Figure 2.5 Standard Configuration, Back View

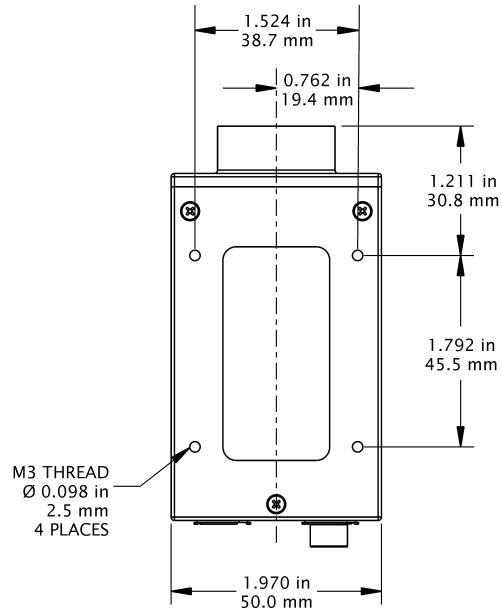


Figure 2.6 Standard Configuration, Bottom View

2.2.2 Right Angle Configuration

Height ....1.38 in /35 mm
Width ....1.97 in/50 mm
Length ....3.16 in/131 mm
Weight ....6.7 oz/190 g (without lens)

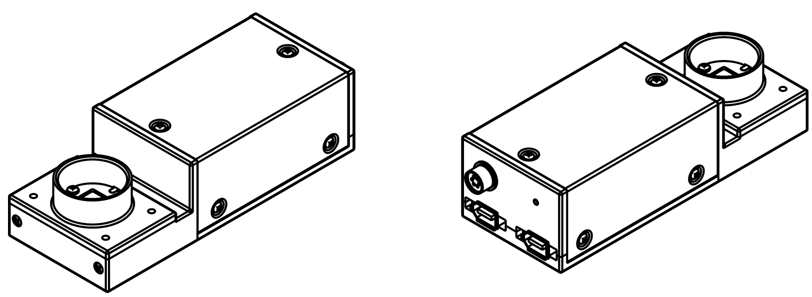


Figure 2.7 Right Angle Configuration

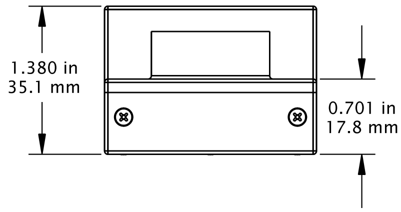


Figure 2.8 Right Angle Configuration, Front View

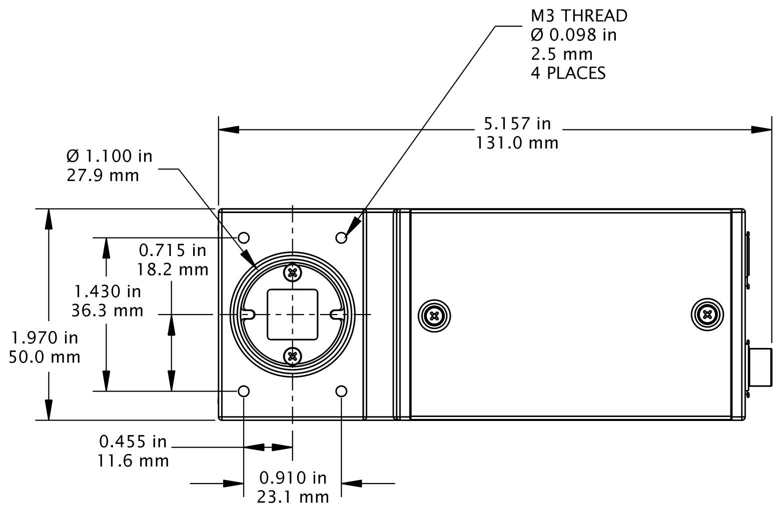


Figure 2.9 Right Angle Configuration, Top View

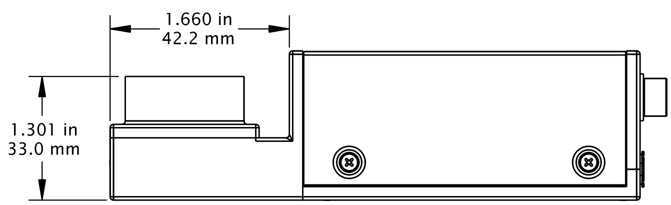


Figure 2.10 Right Angle Configuration, Side View

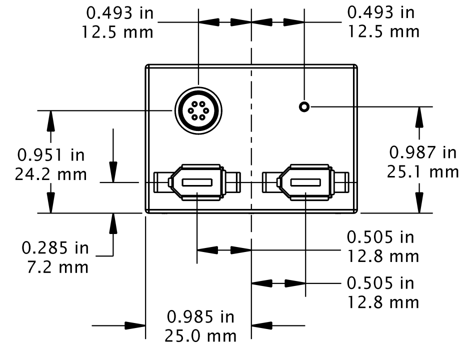


Figure 2.11 Right Angle Configuration, Back View

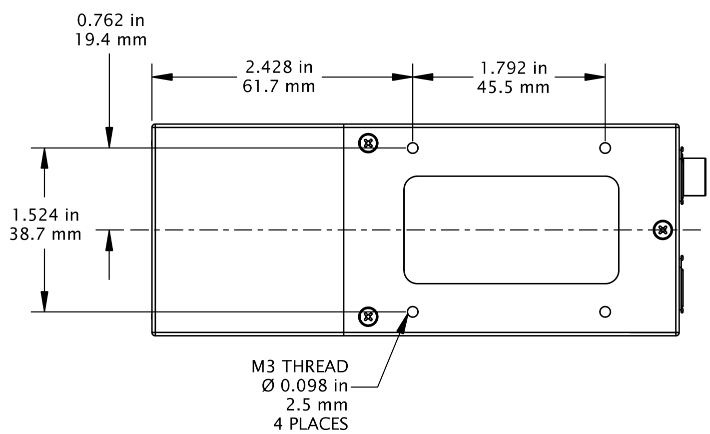


Figure 2.12 Right Angle Configuration, Bottom View

2.3 Handling and Care

Caution:

Do not open the camera housing.

ESD— Handle the PL-A741 as ANSI/ESD S20.20 Class Type 3 (4000 V).

Cleaning

Housing: Clean the housing with a cloth dampened (not dripping) with reagent-grade isopropyl alcohol.

Glass: Clean the protective glass according to the instructions in Section 2.4.2 (on page 18).

2.4 Lens and Protective Glass

2.4.1 Lens

The Camera's C-mount fits a standard C-mount lens (1" × 32 tpi). A lens is not included.

2.4.2 Protective Glass

Each Camera is fitted with a piece of BK7 glass to protect the sensor. The glass is held in place by the Camera's C-mount assembly (Figure 2.13, on page 19) and can be removed if required. (The assembly is identical for the C-mount of the Right Angle configuration.)

The protective glass can be removed for cleaning or for replacement with a special filter. **Because of the risk of damage to the sensor, only the manufacturer should perform modifications to the camera fittings.**

Caution:

Removing the glass increases the likelihood of contaminating the image sensor. If the glass is not place, operate the camera under clean-room conditions only.

Removing the Glass:

1. Ensure that the FireWire cable is disconnected from the Camera. Remove the lens or other C-mount attachment.
2. Remove the screws holding the clamp in place (see Figure 2.13, on page 19).
3. Using a pair of fine-point tweezers, carefully lift out the clamp.
4. Place a lint-free cloth over the C-mount. Slowly tip the Camera until the glass lands gently in the cloth.
5. Store the clamp and screws.
6. Cover the C-mount to protect the sensor from contaminants.

Cleaning the Glass:

Minor cleaning:
1. Carefully apply puffs of dry, compressed air to move particles off the center of the glass.
2. Remove oils with a swab moistened with reagent-grade isopropyl alcohol. The swab should be damp but not dripping.

Major cleaning: Remove the glass as directed above and clean with reagent-grade isopropyl alcohol.

Caution:

<u>Do not</u> use acetone to clean the glass.

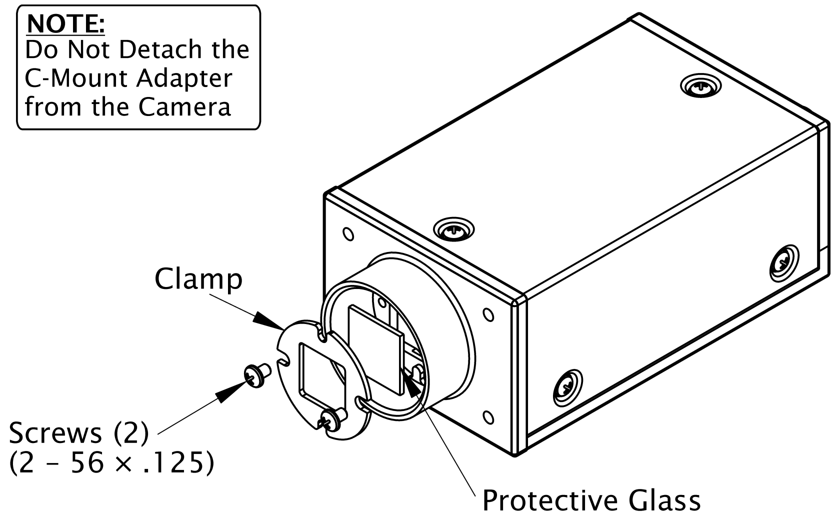


Figure 2.13 Camera C-Mount, Exploded View (Standard Configuration shown)

2.5 Mounting

Bottom Mount

As shown in Figure 2.6 on page 15 (Standard configuration) and Figure 2.12 on page 17 (Right Angle configuration), there are four M3 threaded holes on the bottom of the PL-A741 Camera. These holes can be used to attach the camera to an optional ¼-20 nut mounting plate (available from your PixeLINK vendor) or to mount the camera to a custom fixture.

Sensor-Face Mount

As shown in Figure 2.3 on page 13 (Standard configuration) and Figure 2.9 on page 16 (Right Angle configuration), there are four M3 threaded holes surrounding the C-mount. These holes can be used to mount the Camera to a custom fixture.

Mounting Clearances

Allow sufficient clearances for access to connectors and the status light (back of camera).

The clearances necessary for adequate cooling are dependent on the ambient temperature of the operating environment and the thermal conductivity of the mounting hardware. The camera dissipates 4.2 W, so for optimal performance, the camera should be mounted on a metal plate or fixture. A metal ¼-20 nut mounting plate is available from your PixeLINK vendor.

Allow enough room to keep the camera's internal temperature within tolerance. For temperature ranges, see Section 2.7 (on page 22).

2.6 Connectors

The PL-A741 Camera has the following connectors on the back, as shown in Figure 2.14, (below):

- 2 six-pin FireWire Connectors—Combined power, data, and control signals (see Section 2.6.1, below)
- A six-pin Machine Vision connector—Trigger, strobe, pulse (See Section 2.6.2, on page 21)

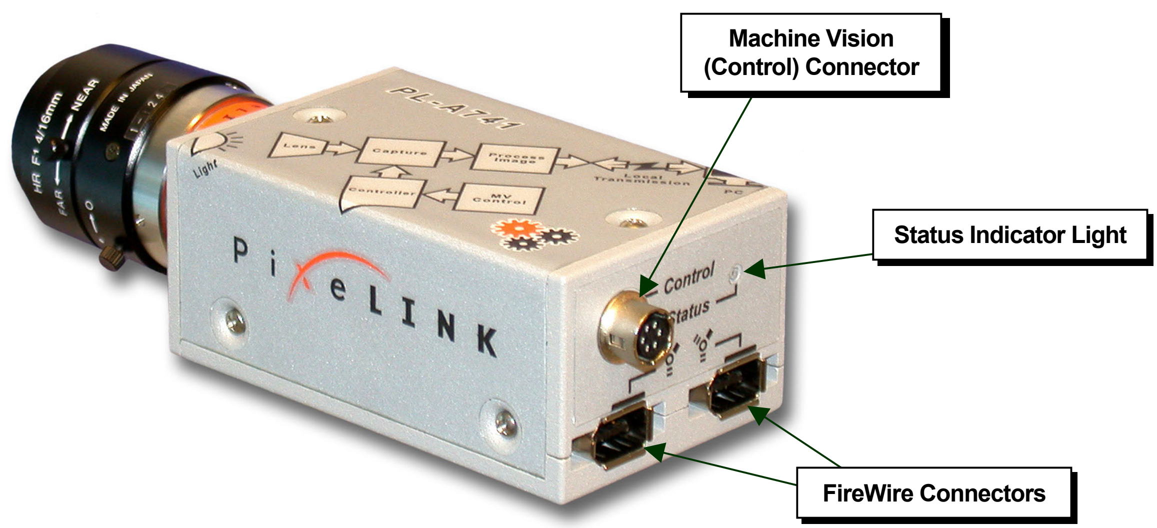


Figure 2.14 Back View of the PL-A741, Showing Connectors

2.6.1 FireWire Connectors (Power, Data, and Control Signals)

Each PixeLINK Camera has two six-pin FireWire connectors (ports), allowing several devices to be daisy-chained. The FireWire cable carries image data, power and control signals.

The connectors are compatible with both regular and harness-type (latching) FireWire plugs.

You do not need to power down the computer to connect or disconnect a camera, nor do you need to reboot after this is done.

Single camera connection: The FireWire connectors are functionally identical. Either one of them can be used for a single connection. Do not make more than one connection between the camera and the computer—the equipment will not be able to communicate properly and the system can crash.

Multiple camera connections: When connecting multiple cameras, do not create a functional loop. The number of cameras that may be managed simultaneously depends on the total bandwidth and may be limited by power availability and the processing capabilities of the host computer. For more information, see Section 2.7.5 (on page 24).

2.6.2 Machine Vision Connector (Trigger, Strobe, Pulse)

The PL-A741 is equipped with a 6-pin Machine Vision connector, as shown in Figure 2.14 (on page 20), for external control of a trigger, strobe or pulse.

The connector pinout is listed in Table 2.1 (on page 22). The connector's interface schematic is shown in Figure 2.15 (below). The mating plug for the connector is an **HR10A-7P-6P**, a 6-pin round plug connector with solder-cup pins for the cable wires.

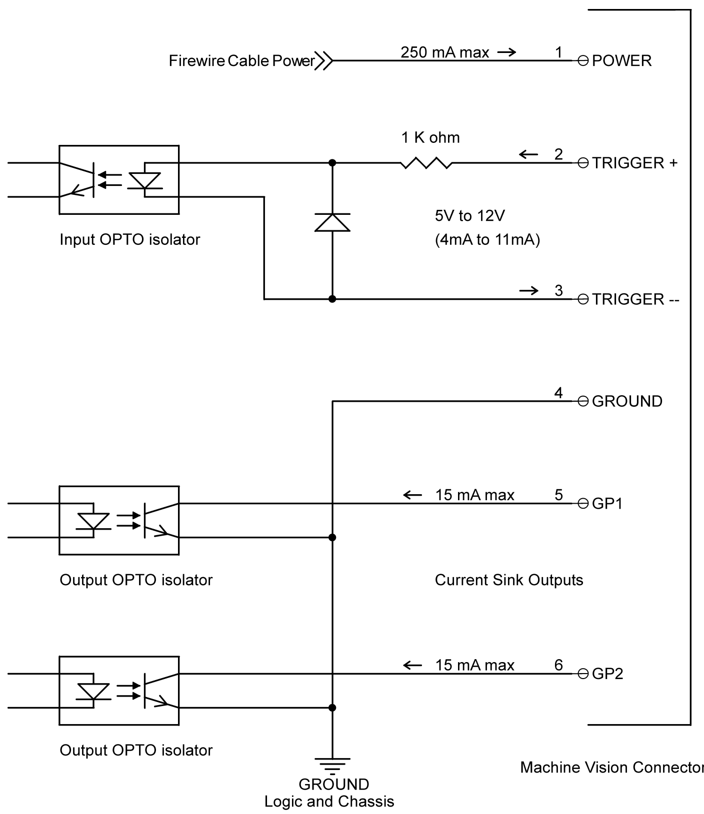
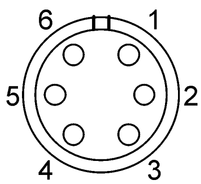


Figure 2.15 Machine Vision Connector—Interface Schematic

Table 2.1 Pinout of the Machine Vision Connector



(Pinout as viewed on the back of the camera, or at the solder-end of the mating connector)

Pin	Pin Name	Function	Comments
1	POWER	FireWire cable power, 8–30 V DC (typically 12 V)	• Maximum current available from this pin is 250 mA.
2	TRIGGER+	Positive terminal of optically isolated trigger input	• To initiate a trigger, apply a voltage of between 5 V and 12 V (4–11 mA) to the trigger terminals. • The optically isolated trigger input circuit includes an internal 1 kΩ series resistor. • When using a voltage higher than 12 V, add an external resistor with adequate power rating. • A typical application of the trigger input is to connect Pin 1 (POWER) to Pin 2 (TRIGGER+) and use the open collector output of an external position sensor device to complete the circuit from Pin 3 (TRIGGER–) to Pin 4 (GROUND). **Timing:** See Section 4.1 (on page 33) **PixeLINK API**--Refer to the functions *PxLSetFeature, PxLGetFeature*
3	TRIGGER–	Negative terminal of optically isolated trigger input	
4	GROUND	Logic and chassis ground	
5	GP1	General Purpose Output (GPO) 1, for strobe, pulse	• Open-collector current sink output from optical isolator. Provides current sink to Pin 4 (GROUND) when GP1/GP2 (as appropriate) is active. • Maximum current is 15 mA. **Timing:** See Section 4.2 (on page 35) **PixeLINK API**--Refer to the functions *PxLSetFeature, PxLGetFeature*
6	GP2	General Purpose Output (GPO) 2, for strobe, pulse	

2.7 Operation

2.7.1 Operating Environment

Table 2.2 Operating Environment Conditions

Condition	Limits
Temperature	0°C to +50°C (32°F to 122°F)
Temperature change	< 10°C/minute (< 18°F/minute)
Shock	50 G
Vibration	10 G at 20-200 Hz
Humidity	20% - 80% non condensing

2.7.2 Storage Environment

Table 2.3 Storage Environment Conditions

Condition	Limits
Temperature	-40°C to + 75°C (-40°F to 167°F)
Temperature change	< 10°C/minute (< 18°F/minute)
Shock	50 G
Vibration	10 G at 20-200 Hz
Humidity	20% - 80% non condensing

2.7.3 Power

The host computer can supply power to the camera via the FireWire bus if the computer has a six-pin FireWire connector and the total power demand on the bus does not exceed the bus capacity.

Other systems—such as laptop computers or those with several FireWire devices daisy-chained—require an external 12 V supply to power the camera. This external power supply must be connected to the FireWire bus. For information about laptop accessory kits that include external power supplies, contact your PixeLINK vendor.

Power Demand

The PL-A741 draws 4.2 watts from the FireWire cable. For a single camera, this is 350 mA @ 12 V. For multiple cameras connected to a single bus, see Power Limitation (below).

Power Limitation

The power capacity of the bus depends on the FireWire adapter card installed in the host computer. To comply with the FireWire specification, the adapter card may supply a maximum of 1.5 A per FireWire port. However, it is permissible for the card's limit to be lower than 1.5 A per port (say, 1.0 A). Note that many cards do not state their current limits.

In a multiple camera system, the current required per camera increases with each additional camera attached to the bus. Each additional camera causes the bus voltage to drop, so the current demand is increased to maintain a constant power draw of 4.2 W per camera.

Hence, most two-port FireWire adapter cards will power only three cameras simultaneously (regardless of whether one or both ports of the card are in use).

If the current demand exceeds the card capacity, the status lights on the cameras will switch off. The status light is located on the back of the camera, as shown in Figure 2.14, (on page 20).

2.7.4 Status Indicator Light

The camera has a status indicator light, located on the back, as shown in Figure 2.5 (on page 14) and in Figure 2.14 (on page 20). This light indicates the operational status of the camera, as described in Table 2.4 (below).

Table 2.4 Status Indicator Signals

Signal (Type and Color)	Status
Solid **GREEN**	The camera is ready for operation.
Flashing **GREEN**	The camera is performing a operation.
Flashing **RED**	The camera is being initialized. **OR** The camera has issued a warning on the latest command received.
Solid **RED**	The camera has experienced an unrecoverable error—contact PixeLINK (see page 49 for contact information).
Off	The computer's FireWire adapter card cannot supply adequate current for the number of devices on the bus. Detach some of the devices or attach a power supply to the FireWire cable.

2.7.5 Connecting Multiple Cameras ("Daisy Chaining")

When connecting multiple cameras, do not create a functional loop. This will have the same consequences as making a duplicate connection between a single camera and the computer. For example, if you connect camera A to camera B and camera B to camera C, you should not connect camera A directly to Camera C.

Power Limitation

Most two-port FireWire adapter cards can power up to three cameras simultaneously. For more information, see Power Limitation (on page 23).

FireWire Bandwidth Limitation

The FireWire bus requires that the sum of the packet sizes of the attached cameras be less than 4800. Packet sizes for cameras operating with 640 × 480 and 1024 × 1280 Regions of Interest (ROI) are given in Table 2.5 (on page 25).

Table 2.5 Packet Sizes

Clock Frequency (MHz)	Region of Interest (ROI)	
	640 × 480	1024 × 1280
4	640	1280
6	1280	1280
8	1280	1280
12	1920	2560
16	3200	2560
24	3840	2560

Example: Based on the FireWire bus only, what is the largest number of cameras that can be managed simultaneously if each one operates with 640 × 480 ROI and an 8 MHz clock frequency?

Solution: The FireWire packet limit is 4800. From Table 2.5 (above), the packet size for a 640 × 480 ROI at 8 MHz is 1280.

$$\frac{\mathbf{4800}}{\mathbf{1280}} = \mathbf{3.75}$$

Therefore, no more than three Cameras can be managed simultaneously at 640 × 480 and 8 MHz.

To confirm:

The total packet size for four cameras is 4 × 1280 = 5120, which exceeds the FireWire packet limit. The total packet size for three cameras is 3 × 1280 = 3840, which is less than the limit.

3

Camera Features

The PL-A741 is compliant with the **IIDC 1394-based Digital Camera Specification** (Version 1.30). The IIDC specification requires that the camera hardware retain a record of all the features supported by the camera. The software and drivers used to control the camera query the camera to retrieve a list of supported features and other information. The specification also allows the definition of advanced features specific to a particular camera model.

This section

- outlines basic features of the PL-A741 camera, and
- provides a detailed description of advanced features specific to the PL-A741.

This information is intended for **system integrators** and **software developers** who are

- creating low-level drivers for the PL-A741, or
- integrating the advanced features of the camera into IIDC applications using third-party software development tools.

Users of the proprietary PixeLINK driver or the PixeLINK Camera API Version 4.0 might find the information in this section to be too low level and detailed. These people should instead refer to the PL-A741 User's Manual and the PixeLINK API Reference Manual.

3.1 Supported Features

Refer to the IIDC specification for more information on accessing and controlling the basic features.

3.1.1 Video Formats

The PL-A741 supports video Format 0 (VGA non-compressed format, 640 × 480 maximum) for backwards compatibility with older IIDC 1.04 drivers and DirectShow. The camera also supports Format 7 (partial image sizes).

3.1.2 Basic Features

The PL-A741 supports basic features as shown in Table 3.1 (on page 27).

Table 3.1 PL-A741 Basic Features

Feature	Supported	Unit	Type	One-Push Auto	Read Out	On/Off	Auto	Manual	Min Value	Max Value
Brightness	Yes	%	Absolute	No	Yes	No	No	Yes	0	100
Shutter	Yes	s	Absolute	No	Yes	No	No	Yes	0.00004	1
Gain	Yes	dB	Relative	No	Yes	No	No	Yes	0	
Temperature	Yes	°C	Absolute	No	Yes	No	No	No		
Trigger	Yes	*See trigger mode descriptions (on page 33)*								

3.1.3 Advanced Features

The PL-A741 provides controls for the following Advanced Features:

Camera information retrieval: This feature reports the camera's hardware and firmware information, specifically the firmware version, the FPGA version, the camera's serial number, and the product ID.

Trigger features: The standard IIDC specification is used whenever possible. However, the standard does not allow the time to be set between a trigger event and the start of integration, so a PixeLINK extension is used to enable this feature. See Section 4.1 (on page 33) for more information about timing.

GPOs/GPIOs: The IIDC specification does not apply to strobes, flashes, or LEDs. A PixeLINK extension allows control of multiple strobes and a general purpose I/O. The extension also allows control of the time between a trigger event and the activation of a strobe, the duration of the strobe and the polarity of the strobe signal. See Section 4.2 (on page 35) for more information about timing.

Extended shutter (Knee points): This feature allows for multiple-slope-exposure captures to enable a larger dynamic range. (See Table 3.2 on page 28.)

Lookup Table (LUT): This feature allows a user-specified lookup table to be applied to the image data, enabling custom filtering and image processing. (See Table 3.2 on page 28.)

Descriptors: A descriptor is a collection of camera feature properties that is applied to a frame. Since each frame can have a unique descriptor, custom descriptors can be used to change camera settings on a frame-by-frame basis for each frame in the video stream.

Table 3.2 Details of Selected Advanced Features

Feature	Supported	Unit	Type
Frame Rate	Yes	fps	Absolute
Trigger Delay	Yes	s	Absolute
Extended Shutter Knee Points	Yes	s	Absolute

3.2 PL-A741 Advanced Features

The Advanced Features of the PL-A741 are controlled using IIDC Configuration Status Registers (CSRs). Table 3.3 (below) lists the CSRs applicable to the PL-A741.

Table 3.3 Advance Feature Configuration Status Registers of the PL-A741

Offset*	Name	Field	Bit	Description
000h 004h	ACR		[0..63]	Advanced Feature Access Control Register (See **NOTE 1**)
CAMERA INFO REGISTERS (READ ONLY)				
008h	SERIAL_OFFSET	Value	[0..31]	Quadlet offset of the Serial Number string from the base address of the initial register space
00Ch	SERIAL_LENGTH	Value	[0..31]	Length in bytes of the Serial Number string
010h	FPGA_VERSION	Value	[0..31]	Fpga Version in format Byte0.Byte1.Byte2.Byte3
014h	FW_VERSION	Value	[0..31]	Firmware Version in format Byte0.Byte1.Byte2.Byte3
018h	CAM_DSC_OFST	Value	[0..31]	Quadlet offset of the Camera Description string from the base address of the initial register space (See **NOTE 2**)
01Ch	CAM_DSC_LNTH	Value	[0..31]	Length in bytes of the Camera Description string
ADVANCED FEATURE INQUIRY REGISTERS (READ ONLY)				
100h	NAME_INQ	Presence_Inq	[0]	Presence of this feature
			[1..15]	Reserved
		Max_Length	[16..31]	Maximum length of Camera Name in bytes
104h	NAME_OFFSET	Value	[0..31]	Quadlet offset of the Camera Name string from the base address of the initial register space (See **NOTE 2**)
108h	NAME_LENGTH	Value	[0..31]	Length in bytes of the Camera Name string (Read Only)

Offset*	Name	Field	Bit	Description
10Ch	DESC_INQ	Presence_Inq	[0]	Presence of this feature
			[1..7]	Reserved
		Max_Num	[8..15]	Maximum number of Descriptors
		Struct_Ver	[16..31]	Version of the Descriptor Structure.
110h	DESC_OFFSET	Value	[0..31]	Quadlet offset of the Descriptor Structure from the base address of the initial register space (See **NOTE 2**)
114h	DECIM_INQ	Presence_Inq	[0]	Presence of this feature
		In_Desc_Inq	[1]	Can this feature have different values in different descriptors 1 = Yes, 0 = No
			[2..7]	Reserved
		Base	[8..15]	Base for decimation. Maximum decimation is Base^Max_Value (except when Base = 1 then it is Max_Value)
		Max_Value	[16..31]	The maximum value for the decimation variable
118h	FRAME_RATE_ABS	Offset	[0..31]	Quadlet offset of the absolute value CSR for the Frame Rate (only valid in Format 7)
11Ch	FLIP_INQ	H_Pres_Inq	[0]	Presence of Horizontal Flip
		In_Desc_Inq	[1]	Can Horizontal Flip have different values in different descriptors 1 = Yes, 0 = No
			[2..15]	Reserved
		V_Pres_Inq	[16]	Presence of Vertical Flip
		In_Desc_Inq	[17]	Can Vertical Flip have different values in different descriptors 1 = Yes, 0 = No
			[18..31]	Reserved
120h	TRIG_ADV_INQ	Presence_Inq	[0]	Presence of this feature
		In_Desc_Inq	[1]	Can this feature have different values in different descriptors 1 = Yes, 0 = No
			[2..7]	Reserved
		Trig_Intern_Inq	[8]	Can the camera be triggered internally (free running) 1 = Yes, 0 = No
		Trig_Type_Inq	[9]	Can the type of trigger be changed 1 = Yes, 0 = No
			[10..31]	Reserved
124h	TRIG_DEL_ABS	Offset	[0..31]	Quadlet offset of the absolute value CSR for Trigger Delay
128h	GPIO_INQ	Presence_Inq	[0]	Presence of this feature
		In_Desc_Inq	[1]	Can this feature have different values in different descriptors 1 = Yes, 0 = No
			[2..3]	Reserved
		Number	[4..7]	Number of GPIO lines (1-15)
		Mode0_Inq	[8]	Presence of GPIO Mode 0
		Mode1_Inq	[9]	Presence of GPIO Mode 1

Offset*	Name	Field	Bit	Description
		Mode2_Inq	[10]	Presence of GPIO Mode 2
		Mode3_Inq	[11]	Presence of GPIO Mode 3
			[12..31]	Reserved
12Ch	GPIO_PARM1_ABS	Offset	[0..31]	Quadlet offset of the absolute value CSR for GPIO 0 Parameter 1 (See **NOTE 3**)
130h	GPIO_PARM2_ABS	Offset	[0..31]	Quadlet offset of the absolute value CSR for GPIO 0 Parameter 2 (See **NOTE 3**)
134h	GPIO_PARM3_ABS	Offset	[0..31]	Quadlet offset of the absolute value CSR for GPIO 0 Parameter 3 (See **NOTE 3**)
138h	EX_SHUTTER_INQ	Presence_Inq	[0]	Presence of this feature
		In_Desc_Inq	[1]	Can this feature have different values in different descriptors 1 = Yes, 0 = No
			[2..3]	Reserved
		Number_Knees	[4..7]	Maximum Number of Knee Points (1-4)
			[8..31]	Reserved
13Ch	XSHUT_KNEE_ABS	Offset	[0..31]	Quadlet offset of the absolute value CSR for Extended Shutter Knee Point 0 (See **NOTE 4**)
140h	LOOKUP_INQ	Presence_Inq	[0]	Presence of this feature
		In_Desc_Inq	[1]	Can this feature have different values in different descriptors 1 = Yes, 0 = No
			[2..7]	Reserved
		Bit_Depth	[8..15]	Bit depth of an entry in the lookup table
		Number	[16..31]	Number of entries in the lookup table
144h	LOOKUP_OFFSET	Offset	[0..31]	Quadlet offset of the first entry in the lookup table. Each entry occupies a whole number of bytes (e.g. a 10bit entry occupies 2 bytes). The table must be written to in order, starting from byte 0.
148h	AUTO_AREA_INQ	Presence_Inq	[0]	Presence of this feature
		In_Desc_Inq	[1]	Can this feature have different values in different descriptors 1 = Yes, 0 = No
			[2..7]	Reserved
		Invalid_Area	[8]	0 = Valid Area, 1 = Invalid Area This feature will be ignored until a valid area is set. The Area must be within the current image region. (See **NOTE 5**)
			[9..31]	Reserved
ADVANCED FEATURE CONTROL REGISTERS				
200h	DESC_CFG	Current	[0..7]	Zero-based index of the descriptor currently in focus (Format 7 only)
		Number	[8..15]	Number of descriptors currently in use (Format 7 only) (See **NOTE 6**)
		Append_Desc	[16]	Append Descriptors to each frame 1 = Append, 0 = Don't Append
			[17..31]	Reserved

PixeLINK

Offset*	Name	Field	Bit	Description
204h	DECIMATION	Value	[0..15]	Value for decimation. Actual decimation is Base^Value except when Base = 1 then it is Value (Format 7 only)
			[16..31]	Reserved
20Ch	TRIGGER_ADV	Source	[0]	Trigger Source 1 = Internal (free running), 0 = External (hardware or software)
		Type	[1]	Trigger Type 1 = Global Shutter, 0 = Rolling Shutter
			[2..31]	Reserved
210h	EX_SHUTTER	Number	[0..7]	Number of Knee Points
			[8..31]	Reserved
214h	LOOKUP_TABLE	On_Off	[0]	Lookup Table On/Off 1 = On, 0 = Off
			[1..31]	Reserved
GPIO CONTROL REGISTERS				
300h	GPIO_0_CFG	On_Off	[0]	GPIO 0 On/Off 1 = On, 0 = Off
		Polarity	[1]	GPIO 0 Polarity 1 = Active High, 0 = Active Low
			[2..23]	Reserved
		Mode	[24..31]	GPIO 0 Mode
304h	GPIO_1_CFG	On_Off	[0]	GPIO 1 On/Off 1 = On, 0 = Off
		Polarity	[1]	GPIO 1 Polarity 1 = Active High, 0 = Active Low
			[2..23]	Reserved
		Mode	[24..31]	GPIO 1 Mode

* Offset from Advanced Feature Offset value (register 0x480)

NOTE 1:

The Feature_Id field of the Advanced Feature Access Control Register is a 48bit value with the following format:

0 – 7	8 – 15	16 – 23	24 – 31	32 – 39	40 - 47
Company_ID (=0x000168)			Advanced_Feature_Unique_ID (=Feature offset)		

For example:
The Feature_Id field for the FLIP_INQ CSR would be 0x000168000114

NOTE 2:

It is not necessary to access the Access Control Register to Read/Write to the value at the offset pointed to by the value in this register.

NOTE 3:

These are the offsets for the absolute value CSRs for the parameters of GPIO 0. The absolute value CSR offsets for GPIO X can be calculated as follows:

quadlet offset of Parameter **N** GPIO **X** = GPIO_PARM**N**_ABS + 3 * **X**.

NOTE 4:

This is the offset for the absolute value CSR for the Extended Shutter Knee Point 0. The absolute value CSR offsets for Knee Point X can be calculated as follows :

quadlet offset of Knee Point **X** = XSHUT_KNEE_ABS + 3 * **X**.

NOTE 5:

In Format 7 the Auto Area must be within the image region specified by the IMAGE_POSITION and IMAGE_SIZE registers. In all other formats the Auto area must be within an image region that has Top and Left coordinates of zero and Height and Width values that correspond to the current Video Format and Mode.

NOTE 6:

Descriptors are added or removed by increasing/decreasing this value. When a descriptor is created it will have the same values as the current descriptor (Current field) or the camera settings if it is the first descriptor to be created. When the descriptor number is decreased then the descriptors with a higher index are removed first.

4

Trigger and GPIO Modes

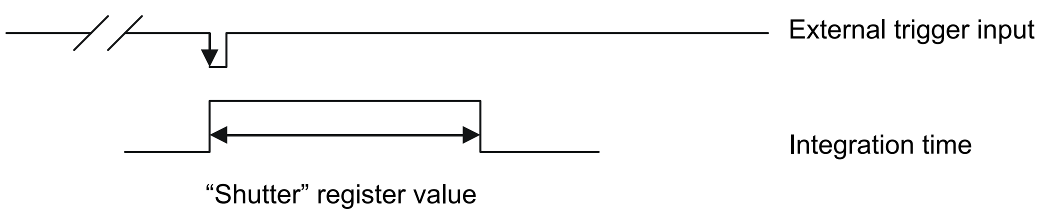
4.1 Triggering

- Trigger Modes 0–3 are defined by the IIDC 1.3 specification
- Trigger Mode 4 is a custom mode.

Note that the trigger input is Low Active.

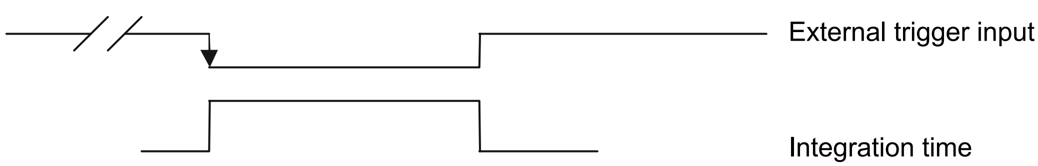
4.1.1 Trigger Mode 0

The camera starts integration of the incoming light from external trigger input falling edge. Integration time is described in "Shutter" register. No parameter is needed.



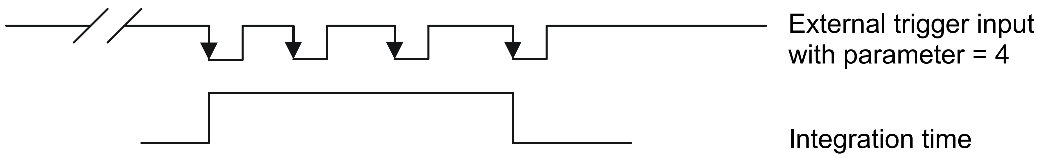
4.1.2 Trigger Mode 1

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. No parameter is needed.



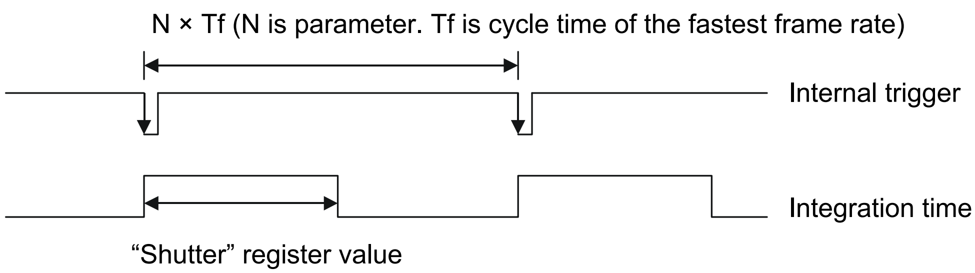
4.1.3 Trigger Mode 2

The camera starts integration of incoming light from first external trigger input falling edge. At the N^{th} (parameter) external trigger input falling edge, integration will be stopped. Parameter is required and must be two or more. (N >= 2)



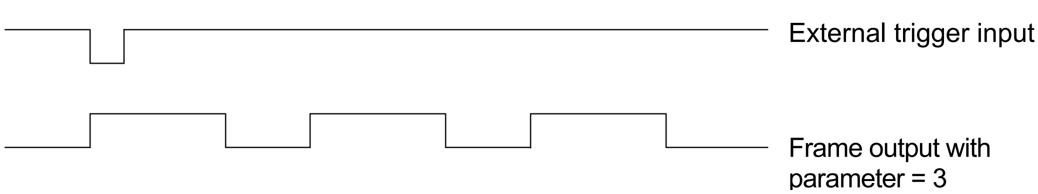
4.1.4 Trigger Mode 3

This is an internal trigger mode. The camera issues trigger internally, and cycle time is N times (parameter) of the cycle time of fastest frame rate. Integration time of incoming light is described in "Shutter" register. Parameter is required and must be one or more. (N >= 1)



4.1.5 Trigger Mode 4

The camera captures N (parameter) frames after a trigger at the specified integration time and frame rate.



4.2 GPIO Modes

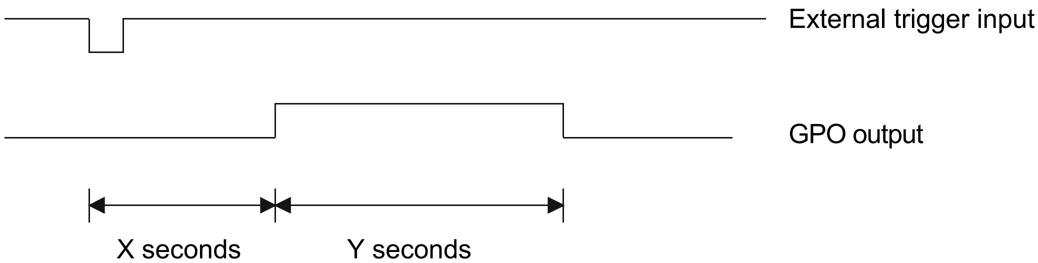
There are four GPIO (General Purpose Input/Output) modes available for use with the GPO pins on the Machine Vision connector (see Section 2.6.2 on page 21).

- Strobe
- Normal
- Pulse
- Busy

Note that the input is Low Active.

4.2.1 GPIO Mode 0 (Strobe)

The GPO is set after a trigger occurs. The GPO pulse occurs X (parameter 1) seconds from the trigger and is Y (parameter 2) seconds in duration.

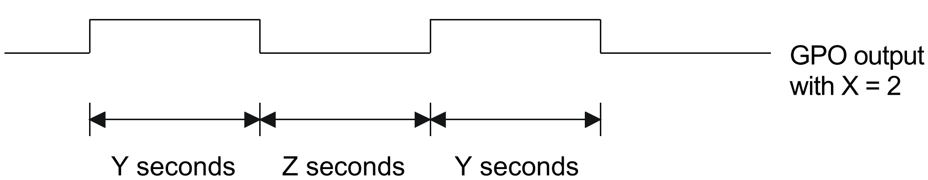


4.2.2 GPIO Mode 1 (Normal)

The GPO is set to either low or high, depending on the value of Polarity.

4.2.3 GPIO Mode 2 (Pulse)

The GPO is pulsed whenever it is switched on. The GPO outputs X (parameter 1) pulses of Y (parameter 2) seconds in length separated by Z (parameter 3) seconds.



4.2.4 GPIO Mode 2 (Busy)

The GPO is set whenever the camera is unable to respond to a trigger. This is only useful when using Trigger Type Hardware or Software; for all other modes it will always be set.

5

Extensions To IIDC 1394-Based Camera Spec v1.30

Table 5.1 Color Coding Inquiry CSR for Format_7

Offset	Name	Field	Bit	Description	
014h	COLOR_CODING_INQ	Mono8	[0]	Y only, Y=8bits, non compressed	ID=0
		4:1:1 YUV8	[1]	4:1:1, Y=U=V=8bits, non compressed	ID=1
		4:2:2 YUV8	[2]	4:2:2, Y=U=V=8bits, non compressed	ID=2
		4:4:4 YUV8	[3]	4:4:4, Y=U=V=8bits, non compressed	ID=3
		RGB8	[4]	R=G=B=8bits, non compressed	ID=4
		Mono16	[5]	Y only, Y=16bits, non compressed	ID=5
		RGB16	[6]	R=G=B=16bits, non compressed	ID=6
		Bayer8	[7]	Bayer Pattern encoding, Red pixel first, 8bits/pixel	ID=7
		Bayer16	[8]	Bayer Pattern encoding, Red pixel first, 16bits/pixel	ID=8
			[9..30]		
		In_Desc_Inq	[31]	Can the Color Coding ID have different values in different descriptors 1 = Yes, 0 = No	

Table 5.2 Feature Element Inquiry CSRs

All feature element inquiry registers (Offset 500h to 5FFh) have the following bit added to them:

Offset	Name	Field	Bit	Description
5XXh	XX_INQ		[0..1]	Same as DCAM Specification
		In_Desc_Inq	[2]	Can this feature have different values in different descriptors 1 = Yes, 0 = No
			[3..31]	Same as DCAM Specification

6

Features That Affect Isochronous Packet Size or Format

This list includes standard DCAM registers as well as extended registers:

Status and Control Registers

CUR_V_FRM_RATE
CUR_V_MODE
CUR_V_FORMAT
ISO_SPEED

Format 7 Registers

IMAGE_SIZE
COLOR_CODING_ID
BYTE_PER_PACKET

Extended Registers

DECIMATION
FRAME_RATE
DESCRIPTOR (structure at DESC_OFFSET)

When any of these registers are written to ISO_EN, ONE_SHOT and MULTI_SHOT should all be set to 0. Otherwise, the written values are ignored and an error is returned.

7

Feature Unit List

Table 7.1 Feature Units

Feature	Unit	Type
Brightness	Percentage	Absolute
Auto Exposure	Exposure Value	Relative
Sharpness	Percentage	Relative
White Balance	Degrees Kelvin	Absolute
Hue	Degrees	Relative
Saturation	Percentage	Relative
Gamma	None	Relative
Shutter	Seconds	Absolute
Gain	dB	Relative
Iris	F Number	Absolute
Focus	Meters	Absolute
Temperature	Degrees Celcius	Absolute
Zoom	Power	Relative
Pan	Degrees	Relative
Tilt	Degrees	Relative
Frame Rate	Frames Per Second	Absolute
Trigger Delay	Seconds	Absolute
Extended Shutter Knee Points	Seconds	Absolute

☐ = Advanced Features

8

Frame Format

A frame coming from the camera has the following format upon reaching the driver:

Byte 0	Byte 1	Byte 2	…	Byte N-2	Byte N-1
Frame Descriptor (See Descriptor Structure Format on page 40)					

= DCAM Frame padding region

N = Number of bytes per frame (width * height * bytes per pixel)

If multiple descriptors are being used, then a synchronization code is encoded in the first four pixels. This code ensures that the driver knows which frame it is receiving from the camera, in case the frame size changes between descriptors. The code is encoded as follows:

Byte 0	Bits 7 … 0	
Byte 1	Bits 7 … 1	Code Bit 7
Byte 2	Bits 7 … 0	
Byte 3	Bits 7 … 1	Code Bit 6
Byte 4	Bits 7 … 0	
Byte 5	Bits 7 … 1	Code Bit 5
Byte 6	Bits 7 … 0	
Byte 7	Bits 7 … 1	Code Bit 4
Byte 8	Bits 7 … 0	
Byte 9	Bits 7 … 1	Code Bit 3
Byte 10	Bits 7 … 0	
Byte 11	Bits 7 … 1	Code Bit 2
Byte 12	Bits 7 … 0	
Byte 13	Bits 7 … 1	Code Bit 1
Byte 14	Bits 7 … 1	
Byte 15	Bits 7 … 1	Code Bit 0

Where "Code" is an 8-bit number (Bit 7 = MSB). Code is zero for the first frame of a video stream, and it increments by one for each frame after that, rolling over once it reaches the number of descriptors currently in use (i.e. if three descriptors are in use Code will roll over when it reaches 2).

9

Descriptor Structure Format

The descriptor structure has two parts:

- The first section of this structure contains a list of all the features that can change between descriptors.
- The second section contains all other features.

The first section of the structure is attached to the end of each frame sent to the driver. The entire structure (first and second sections) is used when accessing the Descriptor structure located at DESC_OFFSET.

The structure is required to be as small as possible, so only features that can change from frame-to-frame with the host computer's "knowledge" are included in the first section (such as features that may be in "Auto" mode). This structure is referred to in the PixeLINK API Reference Version 4.

Three versions of the descriptor structure follow.

9.1 Descriptor Version 0x0000

```
typedef struct _DESCRIPTOR
{
      // There is no data attached to the end of a frame for version
0x0000
#if FRAME_STRUCTURE
} DESCRIPTOR, *PDESCRIPTOR;
#else // !FRAME_STRUCTURE
      // This section contains all settings that don't change without
      // the host's knowledge
      const U16 wSize = 364;        // Size of this structure (in bytes)
      const U16 wVersion = 0x0000;  // Version of this structure

      float fBrightness;            // Brightness
      float fAutoExposure;          // Auto Exposure
      float fSharpness;             // Sharpness
      float fWhiteBalance;          // White Balance
      float fHue;                   // Hue
      float fSaturation;            // Saturation
      float fGamma;                 // Gamma
      float fShutter;               // Shutter
      float fGain;                  // Gain
      float fIris;                  // Iris
      float fFocus;                 // Focus
      float fTemperature;           // Temperature
```

```
        float fTriggerParamter;         // Parameter for Trigger
        float fTriggerDelay;            // Trigger Delay
        float fZoom;                    // Zoom
        float fPan;                     // Pan
        float fTilt;                    // Tilt
        float fOpticalFilter;           // Optical Filter
        float fFrameRate;               // Frame Rate
        float fLeft;                    // ROI Left
        float fTop;                     // ROI Top
        float fWidth;                   // ROI Width
        float fHeight;                  // ROI Height
        float fDecimation;              // Decimation
        float fPixelFormat;             // Pixel Format
        float fKneePoint[4];            // Extended Shutter Knee Point
        float fGpioParameter1[16];      // GPIO Parameter 1
        float fGpioParameter2[16];      // GPIO Parameter 2
        float fGpioParameter3[16];      // GPIO Parameter 3
        float fAutoAreaLeft;            // Auto Area Left
        float fAutoAreaTop;             // Auto Area Top
        float fAutoAreaWidth;           // Auto Area Width
        float fAutoAreaHeight;          // Auto Area Height
        U8 byGpioMode[16];              // GPIO Mode
        U8 byGpioPolarity[16];          // GPIO Polarity
        U8 byTriggerMode;               // Trigger Mode
        U8 byTriggerPolarity;           // Trigger Polarity
        U8 byTriggerType;               // Trigger Type
        U8 byReserved;                  // Padding to quadlet align

} DESCRIPTOR, *PDESCRIPTOR;
#endif // FRAME_STRUCTURE
```

9.2 Descriptor Version 0x0001

```
typedef struct _DESCRIPTOR
{
        const U16 wSize;                // Size of this structure (in bytes)
        const U16 wVersion = 0x0001;    // Version of this structure

        float fBrightness;              // Brightness
        float fAutoExposure;            // Auto Exposure
        float fSharpness;               // Sharpness
        float fWhiteBalance;            // White Balance
        float fHue;                     // Hue
        float fSaturation;              // Saturation
        float fGamma;                   // Gamma
        float fShutter;                 // Shutter
        float fGain;                    // Gain
        float fIris;                    // Iris
        float fFocus;                   // Focus
        float fTemperature;             // Temperature
        float fTriggerParamter;         // Parameter for Trigger
        float fTriggerDelay;            // Trigger Delay
        float fZoom;                    // Zoom
        float fPan;                     // Pan
```

```
	float fTilt;			// Tilt
	float fOpticalFilter;		// Optical Filter
	float fAutoAreaLeft;		// Auto Area Left
	float fAutoAreaTop;		// Auto Area Top
	float fAutoAreaWidth;		// Auto Area Width
	float fAutoAreaHeight;		// Auto Area Height

#if FRAME_STRUCTURE
} DESCRIPTOR, *PDESCRIPTOR;
#else // !FRAME_STRUCTURE
	// This section contains all settings that don't change without
	// the host's knowledge
	float fFrameRate;		// Frame Rate
	float fLeft;			// ROI Left
	float fTop;			// ROI Top
	float fWidth;			// ROI Width
	float fHeight;			// ROI Height
	float fDecimation;		// Decimation
	float fPixelFormat;		// Pixel Format
	float fKneePoint[4];		// Extended Shutter Knee Point
	float fGpioParameter1[16];	// GPIO Parameter 1
	float fGpioParameter2[16];	// GPIO Parameter 2
	float fGpioParameter3[16];	// GPIO Parameter 3
	U8 byGpioMode[16];		// GPIO Mode
	U8 byGpioPolarity[16];		// GPIO Polarity
	U8 byTriggerMode;		// Trigger Mode
	U8 byTriggerPolarity;		// Trigger Polarity
	U8 byTriggerType;		// Trigger Type
	U8 byReserved;			// Padding to quadlet align

} DESCRIPTOR, *PDESCRIPTOR;
#endif // FRAME_STRUCTURE
```

9.3 Descriptor Version 0x0002

```
typedef struct _DESCRIPTOR
{
	const U16 wSize;		// Size of this structure (in
	bytes)
	const U16 wVersion = 0x0002;	// Version of this structure

	float fWhiteBalance;		// White Balance
	float fSaturation;		// Saturation
	float fBrightness;		// Brightness
	float fShutter;			// Shutter
	float fGain;			// Gain
	float fKneePoint[4];		// Extended Shutter Knee Point

#if FRAME_STRUCTURE
} DESCRIPTOR, *PDESCRIPTOR;
#else // !FRAME_STRUCTURE
	// This section contains all settings that don't change without
	// the host's knowledge
	float fAutoExposure;		// Auto Exposure
```

```
        float fSharpness;              // Sharpness
        float fHue;                    // Hue
        float fGamma;                  // Gamma
        float fIris;                   // Iris
        float fFocus;                  // Focus
        float fTemperature;            // Temperature
        float fTriggerParamter;        // Parameter for Trigger
        float fTriggerDelay;           // Trigger Delay
        float fZoom;                   // Zoom
        float fPan;                    // Pan
        float fTilt;                   // Tilt
        float fOpticalFilter;          // Optical Filter
        float fAutoAreaLeft;           // Auto Area Left
        float fAutoAreaTop;            // Auto Area Top
        float fAutoAreaWidth;          // Auto Area Width
        float fAutoAreaHeight;         // Auto Area Height

        float fFrameRate;              // Frame Rate
        float fLeft;                   // ROI Left
        float fTop;                    // ROI Top
        float fWidth;                  // ROI Width
        float fHeight;                 // ROI Height
        float fDecimation;             // Decimation
        float fPixelFormat;            // Pixel Format
        float fGpioParameter1[16];     // GPIO Parameter 1
        float fGpioParameter2[16];     // GPIO Parameter 2
        float fGpioParameter3[16];     // GPIO Parameter 3
        U8 byGpioMode[16];             // GPIO Mode
        U8 byGpioPolarity[16];         // GPIO Polarity
        U8 byTriggerMode;              // Trigger Mode
        U8 byTriggerPolarity;          // Trigger Polarity
        U8 byTriggerType;              // Trigger Type
        U8 byReserved;                 // Padding to quadlet align

} DESCRIPTOR, *PDESCRIPTOR;
#endif // FRAME_STRUCTURE
```

10

Operating (Exposure) Modes

The PL-A741 has two shutter types:

- Rolling Shutter (free-running image capture)—See Section 10.2, below
- Synchronous Shutter (triggered image capture)—See Section 10.3, on page 45

Table 10.1 Capture Methods

Method	Capture Type	Shutter Type	Use of Other Sync. Device Possible?** (e.g., Strobe, Flash)
Free-Running Continuous	Continuous	Rolling	No
Internally Triggered Free-Running Continuous	Continuous	Synchronous	No
Hardware Triggered Frame-on-Demand	Frame-on-Demand	Synchronous	Yes
Software Triggered Frame-on-Demand	Frame-on-Demand	Synchronous	Yes

**** This applies only to the use of the camera with the PixeLINK API or PixeLINK software.**

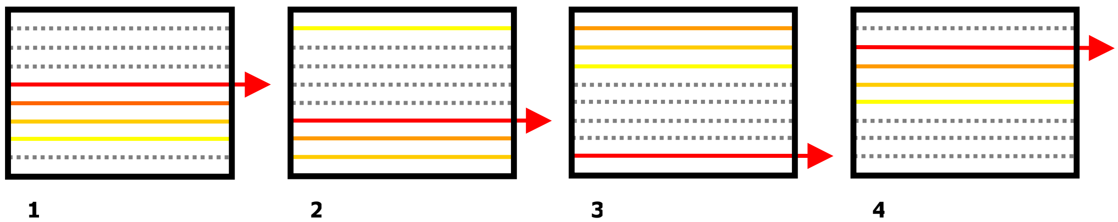
10.2 Rolling Shutter

With a Rolling Shutter, only a few rows of pixels are exposed at one time. The camera builds a frame by reading out the most exposed row of pixels (and ceasing exposure of that row), starting exposure of the next unexposed row down in the **Region of Interest** (**ROI**; the user-specified active area on the imager), then repeating the process on the next most exposed row and continuing until the frame is complete. After the bottom row of the ROI starts its exposure, the process "rolls" to the top row of the ROI to begin exposure of the next frame's pixels.

The exposure down each frame, and from frame-to-frame, remains consistent due to this continuous read-out.

The row read-out rate is constant, so the longer the exposure setting, the greater the number of rows being exposed, or **integrated**, at a given time. ("Integrated" means that the pixels are building up, or integrating, an electrical charge in response to the photons hitting them.) Rows are added to the exposed area one at a time. The more time that a row spends being integrated, the greater the electrical charge built up in the row's pixels and the brighter the output pixels will be. As each fully exposed row is read out, another row is added to the set of rows being integrated.

Example: A very short exposure may be obtained by having only three rows of integration (see Figure 10.1, page 45). This means that as each row is being read out, the three rows ahead of it are being exposed. As each row is read out, another row is added to the group of rows being integrated.



- The arrowhead indicates the row currently being read out.
- Dashed lines indicate inactive rows.
- The three (solid) lines preceding the read-out row indicate the rows being integrated.

Figure 10.1 Rolling Shutter Integration and Read-Out

Frames are kept in a circular buffer—that is, one in which the oldest frame is constantly being overwritten with new frame data—When the camera receives a request for image data, the data is transmitted to the host computer, starting with the oldest frame in the buffer.

Because Rolling Shutter exposes rows in the integration area while reading out fully exposed rows (that is, it does not stop exposure to perform read-out), it provides evenly exposed image data with the greatest possible speed (under the given parameters). Use it when a continuous sequence of frames is required, such as in the capture of smooth video clips.

Each row of pixels has a slightly different exposure start and end times from the adjacent rows, so Rolling Shutter can produce a jagged or blurred effect in fast-action images unless the exposure time is sufficiently short. Under this condition, consider using Synchronous Shutter if the circumstances allow it.

10.3 Synchronous Shutter

With Synchronous Shutter, all rows in the ROI are reset then exposed simultaneously for a specified time. At the end of the exposure time, each pixel value is transferred immediately to an adjacent storage area to await read-out. The pixel values are then read out row-by-row from storage, building the frame. This use of intermediate storage reduces the gradual overexposure that can occur down the image when the rows are read out directly from the active area.

Because all rows are exposed simultaneously, Synchronous Shutter avoids the jagged or blurred affect produced by Rolling Shutter for fast action images. However, because it stops exposure to perform read-out, it does not provide the fastest possible sequence of frames. If speed is the main consideration, use Rolling Shutter if the circumstances allow it.

Synchronous Shutter requires a trigger event to reset the pixel data and start exposure for the entire ROI (as opposed to Rolling Shutter, in which exposure is an on-going process).

See Section 4.1 (on page 33) for information about timing.

Appendix A. Glossary

API Application Programming Interface

DLL Dynamic Link Library

FireWire Apple Computer's trademark for the IEEE 1394 digital bus protocol

GPIO General Purpose Input/Output

GPO General Purpose Output

IEEE Institute of Electrical and Electronics Engineers

fps The frame rate in frames per second.

MB Megabyte

MHz Megahertz

RAM Random Access Memory

RGB Standard for encoding color images (Red, Green, Blue)

Sensor The Camera's image sensing chip

ROI Region of Interest—The portion of the imager area specified for viewing in the preview window or capturing in an image or clip.

Video stream The video image data sent over the FireWire bus (cable) from the camera to the host computer. This is the source of data for the preview window and any captured images or video clips.A

Appendix B. Spectral Response

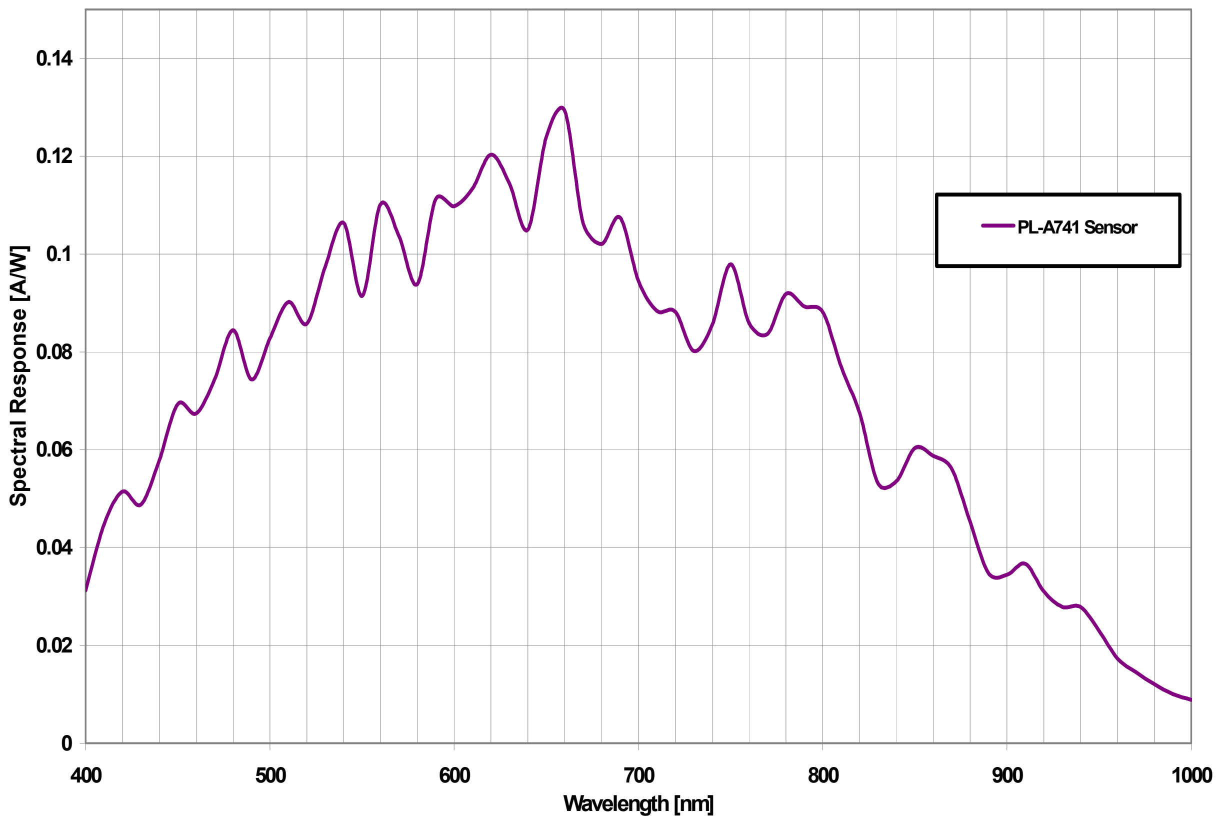


Figure B.1 Spectral Response of the Image Sensor in the PL-A741 Camera

Technical Support

Tech Notes

Tech Notes on a variety of topics are available on our Web site, at http://www.pixelink.com/.

Download the Latest Software and Firmware

The PL-A741's onboard programming (firmware) can be upgraded at your computer. Whether you have a tech support issue or not, we recommend that you check our Web site (http://www.pixelink.com/) regularly for the latest versions of your PixeLINK software and firmware.

Technical Questions

Please follow these steps before contacting our technical support team:

1. Review the relevant sections of the documentation.
2. Check the PixeLINK Web site for
   a. new software and firmware;
   b. applicable Tech Notes.
3. Carefully document the problem you are experiencing, noting any warning or error messages that may appear during operation.
4. Have your Camera's serial number and other identifying information at hand. The serial number can be found on bottom of the Camera.

PixeLINK Customer Support

Email: support@pixelink.com
Web: http://www.pixelink.com/

Index