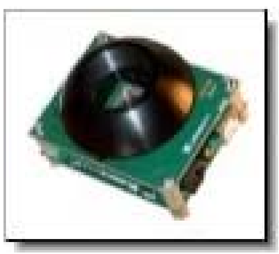
Lumenera corporation

Lumenera USB Camera User's Manual

Release 3.5



Lumenera Corporation
2470 Don Reid Drive, Ottawa, Ontario, Canada. K1H 1E1
Tel: (613) 736-4077 • Fax: (613) 736-4071
www.lumenera.com

License Agreement (Software):

This Agreement states the terms and conditions upon which Lumenera Corporation ("Lumenera") offers to license to you (the "Licensee") the software together with all related documentation and accompanying items including, but not limited to, the executable programs, drivers, libraries, and data files associated with such programs (collectively, the "Software").

The Software is licensed, not sold, to you for use only under the terms of this Agreement.

Lumenera grants to you the right to use all or a portion of this Software provided that the Software is used only in conjunction with Lumenera's family of products.

In using the Software you agree not to:

a) decompile, disassemble, reverse engineer, or otherwise attempt to derive the source code for any Product (except to the extent applicable laws specifically prohibit such restriction);

b) remove or obscure any trademark or copyright notices.

Limited Warranty (Hardware and Software):

ANY USE OF THE SOFTWARE OR HARDWARE IS AT YOUR OWN RISK. THE SOFTWARE IS PROVIDED FOR USE ONLY WITH LUMENERA'S HARDWARE AND OTHER RELATED SOFTWARE. THE SOFTWARE IS PROVIDED FOR USE "AS IS" WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY LAW, LUMENERA DISCLAIMS ALL WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, QUALITY AND FITNESS FOR A PARTICULAR PURPOSE. LUMENERA IS NOT OBLIGATED TO PROVIDE ANY UPDATES OR UPGRADES TO THE SOFTWARE OR ANY RELATED HARDWARE.

Limited Liability (Hardware and Software):

In no event shall Lumenera or its Licensor's be liable for any damages whatsoever (including, without limitation, incidental, direct, indirect, special or consequential damages, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use this Software or related Hardware, including, but not limited to, any of Lumenera's family of products.

Table of Contents

1

Introduction

1.1 The Lumenera USB Camera Family

Lumenera USB Cameras provide a quick and easy means of displaying and capturing high quality video and images on any USB 2.0 equipped desktop, laptop or embedded computer.

Designed with flexibility in mind, each camera model has its own distinct advantage over the others, whether speed, resolution, image quality, sensitivity or price. Because they are USB based, there is no need for a frame grabber. Instead, a single cable provides power, full command and control and data transfer at speeds of up to 24 MB/s.

All cameras have a provision to be externally powered for cases where the USB cable does not supply power (e.g. some USB cards on laptop computers.)

All cameras share the same simple, yet powerful API allowing easy migration from one camera to another. Both board-level and enclosed cameras are available. All cameras also have an optional external interface header for hardware input and output signals and on-board memory for image buffering.

2

Installing and Using the Camera

2.1 Software Installation

The Lumenera USB 2.0 High-Speed camera you have just purchased is designed to operate out of the box with minimal set-up.

Prior to plugging the camera into the computer, you must first install the software.

The software can be found on the CD-ROM that shipped with your product.

2.1.1 Minimum System Requirements

- Windows 2000 (Service Pack 3), or
- Windows XP (Service Pack 1)
- 450 MHz Pentium III or higher (compatible)
- 128 MB RAM
- USB 2.0 Port.

Note: A USB 2.0 Port is required. The camera will not work on a standard USB1.1 port.

2.1.2 Installation Procedure

1. If you are using a 3rd party USB 2.0 PCI add-in card, please ensure the add-in card is properly installed on your computer before proceeding. **We recommend that you use the built-in operating systems USB 2.0 EHCI drivers provided by Microsoft rather than the device drivers supplied with your 3rd party card.**
2. If you have purchased a developers kit and are using the supplied USB 2.0 PCI add-in card, the drivers are built-in when using Win XP with Service Pack 1. Windows 2K users should first upgrade to Service Pack 3, and then go to the Windows upgrade site:

   http://windowsupdate.microsoft.com

   to obtain the USB 2.0 Host controller drivers.

3. **You must be logged into the computer with administrator privileges prior to continuing the installation.**
4. Close all application software that is running and then insert the Lumenera Installer CD into your CD-ROM drive.
5. Wait a few moments for the auto-play function to run the installation program automatically.
6. Follow the onscreen prompts to install the software drivers and user application.
7. After the software has been installed, plug the USB 2.0 Camera into a free USB 2.0 High-Speed port.
8. The Window's New Hardware Wizard will pop-up detecting a new "Lumenera Unconfigured Device". Select **Install the software automatically** from the options that are presented to you and click **Next.** A warning may appear notifying you that the drivers have not been digitally signed by Microsoft. Click **Continue Anyway** to continue with the driver installation. Then click **Finish** to complete the installation of the drivers.
9. After a few seconds the Window's New Hardware Wizard will pop-up again (if it doesn't, disconnect and reconnect the camera to the computer), detecting a "Lumenera USB 2.0 Camera" device. Select **Install the software automatically** from the options that are presented to you and click **Next**. A warning may appear notifying you that the drivers have not been digitally signed by Microsoft. Click **Continue Anyway** to continue with the driver installation. Then click **Finish** to complete the installation of the drivers.

   Note: Windows will require you to re-run steps 8 & 9 each time you plug the camera into a new USB 2.0 port. You must have administrator privileges the first time the camera is used on any given USB 2.0 port. You may wish to repeat the installation steps at this time for all USB 2.0 ports on your computer.

10. Run the application software from your Start menu and enjoy!

2.2 Technical Support

If you have problems installing or using the software or with general camera operation, you can e-mail our technical support group at:

support@lumenera.com

To obtain the latest software release and other technical information you may visit our technical support website at:

www.lumenera.com/tech.html

2.3 Using the Installed Software

All of the necessary software and device drivers are contained in an installation program on the CD-ROM that comes with the camera.

The following files are installed when you run the installation program:

2.3.1 Drivers & INF

Files with a .sys extension are copied to …\SYSTEM32\DRIVERS folder in the standard Windows folder on your system. There are two of these files for each camera model supported by the software. The names of these files are LucamXXX.sys and LuldrXXX.sys (the XXX represents the 3 digit camera ID number.)

Files with a .inf extension are copied to …\INF folder in the standard Windows folder of your system. There are up to two of these files for each camera model supported by the software. The names of these files are LucamXXX.inf and LuldrXXX.inf (the XXX represents the 3 digit camera ID number.)

2.3.2 DirectShow Filters

Several DirectShow related files are installed in the …\SYSTEM folder in the standard Windows folder on your system. These files all have a .ax extension. Their names are:

Lutf.ax

lucustom.ax

lustrcfg.ax

2.3.3 Application Software

The Lucam Capture application (Lucam.exe) is installed in the directory selected during the installation process. The default location is:

C:\Program Files\Lumenera Corporation\Lucam Software

A shortcut to this application is added to the Start Menu at the location selected during installation. The default location is:

Start > Programs > Lumenera > LuCam > Lucam.exe

2.3.4 Development Software (SDK)

If you purchased the Developer's Kit, the Lucam Capture application source code, and the API libraries are installed in folders called "Sample Code" and "SDK", which are in the directory selected during the installation process. The default location is:

C:\Program Files\Lumenera Corporation\Lucam Software

The source code consists of a complete Microsoft Visual C++ 6.0 project. The libraries are also compatible with Visual Basic and Borland C++ Builder.

2.3.5 Documentation

Documentation consisting of this User's Manual, the API reference manual (if you purchased the Developer's Kit) and the latest available Application Notes and White Papers, is installed in a folder called "Documentation" in the directory selected during the installation process. The default location is:

C:\Program Files\Lumenera Corporation\Lucam Software

To obtain the latest documentation and other technical information you may visit our technical support website at:

www.lumenera.com/tech.html

2.4 Using Lucam Capture

The Lucam Capture application is, for the most part, simple to use and self-explanatory. The application makes use of the SDK and provides an example of what the API can do, however, it doesn't make use of all the available features of the API. The complete source code for this application is available to those that purchased the Developer's Kit.

Only one camera may be controlled by each instance of Lucam Capture, but several instances of the application may be run simultaneously. If more than one camera is detected by the application, it pops up a list of their serial numbers, allowing the user to select the camera they wish to control.

2.4.1 Menu Items

"Preview Frame Rate…" will display the average frame rate of the preview window. The average is computed over the whole time span that the display has been actively previewing since the last time Start Preview was pressed.

"Read/Write Registers…" will pop up a dialog allowing you to read and write the registers of the camera. This is an advanced function and should not be used without the advice of our technical support staff.

"Light/Source" provides the option of selecting the ambient lighting source that is being used so that the proper colour correction can be performed by the camera. These options have little effect for some cameras.

"Enable 16-bit Mode" will put the camera into 16-bit mode. You cannot preview data in this mode; but the data will still stream to the computer (see Camera Modes section below for more information about streaming) allowing you to capture 16-bit video frames. (The number of actual valid data bits per pixel will vary by camera model.)

"Monochrome Preview" puts the camera into monochrome mode.

"Image Averaging" averages 5 frames of video together to reduce random image noise, when the Capture button is pressed. This option should not be used when imaging a moving object.

"Image Summing" sums 5 frames of video together to produce a brighter image, when the Capture button is pressed. This option should not be used when imaging a moving object.

"Sharpen Captured Image" applies a sharpening algorithm to the image when it's captured (not the live preview). If an image is currently being displayed, toggling this option will immediately toggle the image in the image display window between sharpened and unsharpened.

"Hue/Saturation…" pops up a dialog that allows you to adjust the hue and saturation of the live preview.

"Display Video Properties…" pops up a "canned" dialog generated by the Lucam API that allows you to adjust video properties (Exposure, Gain, Gamma, Brigtness, Contrast).

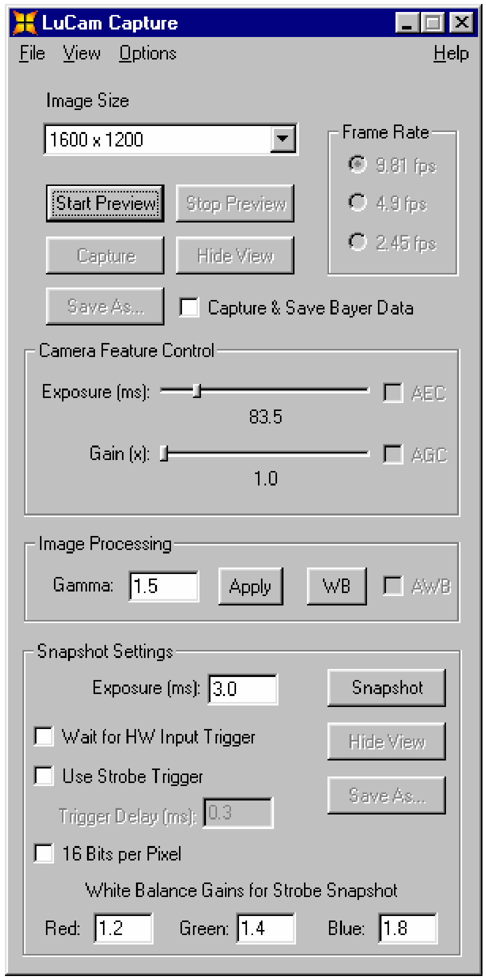


Figure 1 - Lucam Capture Main Window

2.4.2 Buttons and Interface Controls

The "**Start Preview**" button is used to start the video display to the screen.

The "**Stop Preview**" button is used to stop the video display to the screen.

Video Frame Capture

The "**Capture**" button is used to grab a frame of video from the video stream and display it on screen.

The "**Save As…**" button is used to save the image to disk in one of the available formats.

The "**Hide View**" button will close the image display window.

The "**Capture & Save Bayer Data**" toggle button allows you to view and save the raw Bayer data that comes from the camera, before it is processed into 24-bit RGB data. (Color cameras only.) If an image is currently being displayed (with the Capture button), toggling this button will immediately toggle the image in the image display window between Bayer and 24-bit data.

Video Image Control

The "**Image Size**" dropdown list provides the available video display resolutions.

The "**Frame Rate**" toggle buttons provide the selection of the available display frame rates. Not all cameras have this capability.

The "**Exposure**" slider is used to adjust the video exposure time in milliseconds.

The "**AEC**" toggle button is used to toggle the Automatic Exposure Control (not available for all cameras). When selected, the slider changes to "Luminance Target" allowing you to select the average brightness that you want to maintain as ambient lighting changes. The exposure will be automatically adjusted in an attempt to maintain the average brightness.

The "**Gain**" slider is used to adjust the global gain of the camera for both video mode and when using the Snapshot mode (described below). The gain value is a multiplicative factor, so a value of 1 means no gain. When the gain is increased, noise will be introduced in the camera and the image will be degraded. The higher the gain, the more noticeable this is.

The "**AGC**" toggle button is used to toggle the Automatic Gain Control (not available for all cameras). When selected, the slider changes to "Luminance Target" allowing you to select the average brightness that you want to maintain as ambient lighting changes. The gain will be automatically adjusted in an attempt to maintain the average brightness.

> **Note:** When both AEC and AGC are selected, if an increase in brightness is required, exposure is adjusted up first until its limit is reached and then gain is adjusted. When a decrease in brightness is required, gain is adjusted down first until its limit is reached and then exposure is adjusted. This maintains the best image quality.

The "**Gamma**" value is applied to the image to make it look better on screen. It is used to correct the non-linearity inherent in most monitors. A value of 1 represents no gamma correction. Values less than one will make the image appear darker and greater than one will make the image appear brighter. For more information about Gamma and why it's used, consult the following reference: www.poynton.com/GammaFAQ.html

The "**WB**" button performs a color balance (a.k.a. white balance) of the video preview, based on the overall image, using the gray world algorithm. It is done in software by grabbing a video frame, analyzing it, adjusting the color gains and repeating until the colors in the image are balanced. That is, there is an equal amount of Red, Blue and Green in the image. It is best to put a neutral target (e.g. white paper) in front of the camera before performing a color balance.

Snapshot Settings

The "**Exposure**" value is the length of time to expose the snapshot for in milliseconds.

The "**Snapshot**" button is used to grab an image from the camera using its snapshot mode and half-global or global shutter (if available), and display it on screen. (see Shutter Types and Camera Modes sections below for more information about snapshot mode and global shutter)

The "**Hide View**" button will close the snapshot image display window.

The "**Wait for HW Input Trigger**" toggle is used to specify that the snapshot should be hardware triggered using the trigger input of the camera's external header. With this option selected, when the "Snapshot" button is pressed, the software will pause as the camera waits for the hardware trigger before returning the image. There is a built-in time-out of 25 seconds after which time if the hardware trigger has not occurred, the software will resume operation.

The "**Use Strobe Trigger**" toggle is used to specify that during the snapshot exposure, the strobe trigger output should be fired.

The "**Trigger Delay**" value indicates the time in milliseconds between the start of exposure and the rising edge of the strobe trigger pulse.

The "**Save As…**" button is used to save the snapshot image to disk in one of the available formats.

The "**16 Bits per Pixel**" toggles the camera between 8 and 16-bit data mode for snapshot capture.

The "**White Balance Gains for Strobe Snapshot**" values allow you to set the Red, Green and Blue gains to be used during the snapshot capture. This allows you to white balance according to the strobe lighting that is being used. They are only applied if the "Use Strobe Trigger" option is selected.

3

Understanding Your Camera

3.1 Shutter Types

Depending on which camera model you have, the following electronic shutter types may or may not be present. Check the table at the end of this section to determine which camera model has which shutter type. These types are selectable for the snapshot mode of the camera (described in a later section).

3.1.1 Rolling Shutter

With a rolling shutter the exposure process begins, whereby, rows of pixels in the image sensor start exposing in sequence, starting at the top of the image and proceeding row by row down to the bottom. At some later point in time, the readout process begins, whereby, rows of pixels are read out in sequence, starting at the top of the image and proceeding row by row down to the bottom in exactly the same manner and at the same speed as the exposure process.

The time delay between a row starting to expose and a row being read out is the integration time, also known as the exposure time. This integration time can be varied from a single line (start exposure followed by a read out while the next line is exposing) up to a full frame time (last line starts exposing at the bottom of the image before reading starts at the top). In some cases, longer exposures can be obtained by delaying the read out even longer (during which time, the entire array is exposing).

Since the integration process moves through the image over some length of time, skewing of moving objects may become apparent. For example, if a vehicle is moving through the image during capture, light from the top of the vehicle will be integrated at some earlier time than light from the bottom of the vehicle, causing the bottom of the vehicle to appear slanted forward in the direction of motion. For most slow moving objects or still image capture, this motion artifact is not noticeable.

3.1.2 Half Global Shutter

With a half global shutter, the entire image array starts exposing at the same time (globally). At some later point in time, the readout process begins, whereby;

rows of pixels are read out in sequence, starting at the top of the image and proceeding row by row down to the bottom (exactly like the rolling shutter case).

The time between the global start of integration and the start of readout is defined as the exposure time. However, since during readout of the image, the lines are still integrating (like rolling shutter), the actual image exposure differs from the top to the bottom. The difference is the time taken to readout the image and varies for each camera (70 ms is typical). Under bright ambient lighting conditions, the image will appear brighter; the further down the image you go.

Because integration continues to occur during readout, the skewing motion artifact can still occur.

3.1.3 Global Shutter

With a global shutter, the entire image array starts exposing at the same time (globally). At some later point in time, the entire image array stops exposing at the same time and the image is read out in sequence, starting at the top of the image and proceeding row by row down to the bottom (sometimes odd rows are read out first followed by the even rows). The difference from the other modes is that during readout, the imager is no longer integrating light.

The time delay between the start of exposure and end of exposure is defined as the exposure time and it represents the total amount of time that the image integrates.

Because all the pixels start exposure at the same time, integrate over the same interval, and stop exposing at the same time, there is no potential for motion artifacts as there is in the other modes.

Table 1 - Shutter Types by Camera Model

Camera Model	Rolling Shutter	Half Global Shutter	Global Shutter
LU-050, LU-055	Yes	No	Yes
LU-070, LU-075	No	No	Yes
LU-080, LU-085	No	No	Yes
LU-100, LU-105	Yes	Yes	No
LU-110, LU-115	Yes	No	No
LU-120, LU-125	Yes	No	Yes
LU-130, LU-135	No	No	Yes
LU-160, LU-165	No	No	Yes
LU-170, LU-175	Yes	No	No
LU-200, LU-205	Yes	Yes	No
LU-270, LU-275	Yes	No	No
LU-330, LU-335	No	No	Yes

LU-370, LU-375	Yes	No	No
LU-620, LU-625	Yes	No	No

3.2 Scanning Mode

Depending on which model of camera you have, the frame integration will be either progressive scan or interlaced. Check the table at the end of this section to determine which camera model has which scan type.

3.2.1 Progressive Scan

In a progressive scan camera, the entire image is integrated (exposed) at one point in time (for global shutters) or line-by-line from top to bottom (for rolling shutters).

3.2.2 Interlaced Scan

In an interlaced scan camera, the entire image is made up of two fields. Each field is made up of the odd lines of the image (odd field) or the even lines of the image (even field). Each field is captured in a progressive manner (using a global shutter), but the exposure for the second field is started after the first one is read out.

When there is no movement of the object being viewed, you will not see a difference between progressive and interlaced scan images. However, when there is movement of the object, the interlaced scan image will exhibit image artefacts known as the "comb" effect where the edges of the object look like the teeth of a comb because the object is in a different place for the odd versus the even rows of the image.

Table 2 - Scan Mode by Camera Model

Camera Model	Scan Mode
LU-050, LU-055	Progressive
LU-070, LU-075	Progressive
LU-080, LU-085	Progressive
LU-100, LU-105	Progressive
LU-110, LU-115	Progressive
LU-120, LU-125	Progressive
LU-130, LU-135	Progressive
LU-160, LU-165	Progressive
LU-170, LU-175	Progressive

LU-200, LU-205	Progressive
LU-270, LU-275	Progressive
LU-330, LU-335	Interlaced
LU-370, LU-375	Progressive
LU-620, LU-625	Progressive

3.3 Use of Flash or Strobe

A flash or strobe may be used with any camera model and the option is available to provide a programmable trigger signal from the camera to the flash or strobe device to tell it when to fire. However, the type of shutter mode being used will dictate what conditions will be required and how well flash photography will work with the camera.

3.3.1 Flash with Rolling Shutter

The use of a flash with rolling shutter is only feasible for cameras that allow exposures longer than frame read out time (typically about 70 ms). This is because with exposures less than that, only a band across the imager is being exposed at the same point in time and when the flash occurs, it will only illuminate that region of the imager. The flash must be fired at the time when all the pixels of the imager are simultaneously sensitive to light. The strobe signal from the camera is generated at a user selectable delay from that point in time.

Generally, the ambient lighting should be low enough (i.e. dark) so that during the overall exposure the ambient light will not contribute much to the overall brightness of the image. This is particularly true if the flash is being used to "stop the motion" of a fast-moving object, otherwise, blurring or skewing may occur. For imaging still objects, this is not as much of a concern. In this case, you only need to ensure that you are not overexposing the object with both a long exposure and a flash.

3.3.2 Flash with Half Global Shutter

The use of a flash or strobe with an imager using a half global shutter is similar to the rolling shutter case. However, because the imager starts at once exposing all the pixels globally, the strobe signal from the camera is generated at a user selectable delay from the start of exposure. It doesn't have to first wait for the rolling shutter to "open up" all the way, like for rolling shutter mode.

Again, the ambient lighting should be low enough so that during the image read out where the imager is still sensitive, the ambient light will not contribute much to the overall brightness of the image. This is a concern for both moving objects where both blurring and skewing may occur, and still objects where you may have uneven brightness from the top of the image to the bottom (as described in the previous section.)

3.3.3 Flash with Global Shutter

The use of a flash or strobe with a global shutter has no limitations or concerns. The strobe signal from the camera is generated at a user selectable delay from the start of the exposure. Very short, global exposures can be used, so, there will be no blurring or skewing or overexposure due to long exposures.

3.4 Camera Modes

The camera has two operating modes: Streaming Video, and Snapshot.

3.4.1 Streaming Video

In streaming video mode, image frames are continuously being sent from the camera to the computer where they are available for use. The data is pushed from the camera, with no user intervention required. The rolling shutter is always used in this mode. An output signal is provided on the external header indicating the start of exposure for each video frame and can be used to help synchronize events with the video images. The camera will operate with the fastest frame rates in this mode.

3.4.2 Snapshot (Asynchronous Trigger)

Snapshot mode is used to capture one (or more) individual frames in an asynchronous manner. In this mode, the user must initiate the action to start the image retrieval through either hardware or software.

The software trigger is provided using API function calls. The function call is made causing the snapshot to be taken and a single image is returned.

The hardware input trigger (with a user programmable delay) can be used to initiate the snapshot via the external I/O interface. An API function call is made that puts the camera into this "wait for hardware trigger" state and then "blocks" until the hardware trigger is received. Once the trigger is received (or the user selected timeout occurs), the API function returns and passes back the image (or a timeout error code).

Any of the available shutter types can be used with snapshot mode. An output strobe signal with programmable delay can also be synchronized with each snapshot. This is described in more detail in the "External I/O Interface" section below.

Because the snapshot mode is initiated for each image that is desired and images are buffered inside the camera, the amount of time it takes to grab a single snapshot is about twice the time for each streaming video image. Therefore, the effective frame rate is about half of the maximum video frame rate.

3.5 Data Format

Data from the camera can be retrieved in one of two pixel formats. These formats represent the bit depth in bits per pixel [bpp]. Either 8 bpp or 16 bpp can be selected. For 16 bpp, not all of the bits are necessarily valid data bits. Depending on the camera model, 10, 12 or 14 bits will be valid data, with the other 6, 4, or 2 bits always zero. A completely dark pixel will have all valid bits set to zero and a completely light-saturated pixel will have all valid bits set to one. The valid data bits are stored most significant bit aligned in each word and the words are in Little Endian byte order (least significant byte is first of each byte pair). The following table illustrates this point where the data for the first three pixels (completely light-saturated) of an image are represented.

Table 3 - Pixel Data Format for 16 bpp (10 valid data bits)

Pixel	Pixel 1		Pixel 2		Pixel 3	
16-bit Word	Word 1		Word 2		Word 3	
Byte Order	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
	LSB	MSB	LSB	MSB	LSB	MSB
Binary value	11000000	11111111	11000000	11111111	11000000	11111111
Hex value	0xC0	0xFF	0xC0	0xFF	0xC0	0xFF
Decimal value	192	255	192	255	192	255

For monochrome cameras, each byte (8bpp) or word (16bpp) represents one complete pixel in the image.

For color cameras, the data arrives from the camera in the raw Bayer format. The imager in a color camera is a monochrome imager that has a Red, Green or Blue color filter over each pixel. The arrangement of this color filter mosaic is called the Bayer format. An example of this can be seen in Figure 1.

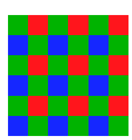


Figure 2 - Example of a 6x6 Pixel Area of Color Imager Mosaic Pattern

Each byte (8bpp) or word (16bpp) will be one of the three mosaic colors: Red, Green or Blue. The order of these colors is camera model dependent and can be found in the following Table.

Table 4 - Bayer Data Color Mosaic Order

Camera Model	Mosaic Order			
	Pixel 1 Row 1	Pixel 2 Row 1	Pixel 1 Row 2	Pixel 2 Row 2
LU-050, LU055	Red	Green 1	Green 2	Blue
LU-070, LU-075	Red	Green 1	Green 1	Blue
LU-080, LU-085	Green 1	Red	Blue	Green 2
LU-100, LU-105	Blue	Green 1	Green 2	Red
LU-110, LU-115	Green 1	Red	Blue	Green 2
LU-120, LU-125	Green 1	Blue	Red	Green 2
LU-130, LU-135	Red	Green 1	Green 2	Blue
LU-160, LU-165	Red	Green 1	Green 2	Blue
LU-170, LU-175	Green 1	Red	Blue	Green 2
LU-200, LU-205	Blue	Green 1	Green 2	Red
LU-270, LU-275	Green 1	Red	Blue	Green 2
LU-330, LU335	Red	Green 1	Green 2	Blue
LU-370, LU-375	Green 1	Red	Blue	Green 2
LU-620, LU-625	Red	Green 1	Green 1	Blue

When using the Lucam Capture application to preview video from a color camera or save images to disk, conversion of the data to standard 24-bit RGB data is done by the software automatically.

When using the API (available with the Developer's Kit), you have complete control over this conversion process.

3.6 Subwindowing, Subsampling & Binning

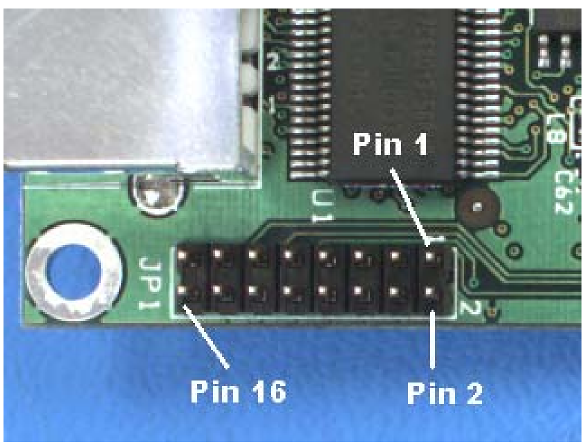
Subwindowing, also known as region of interest, is the ability of the camera to output a smaller image size (subwindow) than the whole imager array. An imager that supports a maximum resolution of 1280x1024 pixels for example, could output a subwindow of 640x480 pixels with the subwindow being positioned nearly anywhere inside the 1280x1024. The subwindow is actually a smaller field of view than the maximum resolution available. There are limitations on the granularity of the subwindow size and on its position within the whole array. The granularity is 8 pixels.

Subsampling, also known as decimation, is the throwing away of every n^{th} pixel or pixel pair in the image in the X and/or Y directions. For example, an imager with a maximum resolution of 1280x1024 could throw away every second pixel in both the X and Y directions and output an image that is 640x512 pixels, yet covers the same field of view of the original full resolution. Not all cameras support Subsampling. Those that do may support subsample levels of 2, 4 or 8. Some cameras even allow different Subsampling in the X vs. the Y directions.

Binning is similar to Subsampling, except instead of throwing pixels away pixels are combined in some fashion. They can be either summed (to provide greater sensitivity) or averaged (to reduce noise). The resulting resolution would be the same as for Subsampling, but all the data would be used. Several cameras support binning with binning levels up to 8 by 8.

3.7 External I/O Interface

For board-level cameras, the External Interface Header can be found in the corner of the PCB next to the silver USB connector. For enclosed cameras, it is found on the side of the camera near the USB connector. It is a male, 2 mm pitch, 16-pin (2 x 8) header. The pin numbering can be seen in Figure 2.



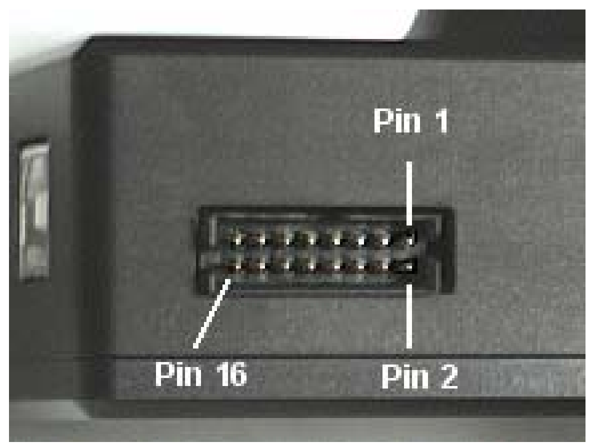


Figure 3 - External Header Location and Pin Numbering

3.7.1 Recommended Mating Connectors

The following mating connectors have been tested to work with the cameras. All of them are for 16-pin (2 x 8), 2mm pitch headers.

- AMP/Tyco P/N 111623-3 IDC Ribbon Cable Receptacle
- Molex GC/Waldom P/N 87568-1661 IDC Ribbon Cable Receptacle
- Molex GC/Waldom P/N 87568-1691 IDC Ribbon Cable Receptacle Locking
- Molex GC/Waldom 51110-1650 Wire Crimp Receptacle
    - Female Crimp Terminal for above – P/N 50394-8100

- Norcomp P/N 2064-16-01-P2 Vertical Dual Row Receptacle
- Sullins P/N PPWN082AFCN Vertical Dual Row Receptacle

All of these connectors can be purchased from Digi-Key® (www.digikey.com) but other parts suppliers may also carry them.

3.7.2 Header Pin-out

Table 5 - Header Pin-out Definition

SIGNAL	PIN #	PIN #	SIGNAL
GPO1 / Strobe Out (AL)	1	2	GND
GPO2 / Strobe Out (AH)	3	4	GND
GPO3	5	6	GND
GPO4 / Video SOF	7	8	GND
GPI1 / Trigger In	9	10	GND
GPI2	11	12	GND
GPI3	13	14	GND
GPI4	15	16	GND

None of the signals can supply much current. Maximum current draw should be kept to less than 5 mA.

3.7.3 Signal Definitions

GPO1 / Strobe Out: Pin 1, LVTTL output (V_{oh} ~ 3.0V, V_{ol} ~ 0V). This signal can be toggled using the LucamGPIOWrite function of the API.

This signal serves double duty and is also used to provide an ACTIVE LOW, 5.5 ms pulse (suitable for triggering a strobe unit) when any of the "Take Snapshot" API functions are used with the "useStrobe" option enabled.   This strobe pulse can be delayed with respect to the start of frame exposure by a user selectable amount (see "Lumenera API Reference Manual" for further details).

GPO2 / Strobe Out: Pin 3, LVTTL output (V_{oh} ~ 3.0V, V_{ol} ~ 0V). This signal can be toggled using the LucamGPIOWrite function of the API.

This signal serves double duty and is also used to provide an ACTIVE HIGH, 5.5 ms pulse (suitable for triggering a strobe unit) when any of the "Take Snapshot" API functions are used with the "useStrobe" option enabled.   This strobe pulse can be delayed with respect to the start of frame exposure by a user selectable amount (see "Lumenera API Reference Manual" for further details).

GPO3: Pin 5, LVTTL output (V_{oh} ~ 3.0V, V_{ol} ~ 0V). This signal can be toggled using the LucamGPIOWrite API function.

GPO4 / Video SOF: Pin 7, LVTTL output (V_{oh} ~ 3.0V, V_{ol} ~ 0V). This signal can be toggled using the LucamGPIOWrite API function.

This signal serves double duty and is also used to provide an ACTIVE HIGH, 85 us pulse each time a frame is output in video mode. The LucamGpoSelect API function is used to enable/disable the Video SOF signal.

GPI1 / Trigger In: Pin 9, LVTTL input (V_{in} min = 0V, V_{in} max = 3.3V). This signal is floating and MUST be driven at all times when being used. The signal status can be obtained by using the LucamGPIORead API function.

This signal serves double duty and is also used to receive an ACTIVE HIGH, LVTTL input (V_{in} min = 0V, V_{in} max = 3.3V) pulse which will trigger the taking of a snapshot, when any of the "Take Snapshot" API functions are used with the "useHwTrigger" option enabled. The active high pulse must have a minimum width of 0.5 us. There is no maximum limit to the trigger pulse width.

GPI2: Pin 11, LVTTL input (V_{in} min = 0V, V_{in} max = 3.3V). This signal is floating and MUST be driven at all times when being used. The signal status can be obtained by using the LucamGPIORead API function.

GPI3: Pin 13, LVTTL input (V_{in} min = 0V, V_{in} max = 3.3V). This signal is floating and MUST be driven at all times when being used. The signal status can be obtained by using the LucamGPIORead API function.

GPI4: Pin 15, LVTTL input (V_{in} min = 0V, V_{in} max = 3.3V). This signal is floating and MUST be driven at all times when being used. The signal status can be obtained by using the LucamGPIORead API function.

3.7.4 Taking a Single-Frame Snapshot with the Camera

The Lumenera API makes use of several of the External Interface Header pins automatically, when the "Take Snapshot" related functions (those that use the LUCAM_SNAPSHOT structure) are called with certain options (see the API documentation for more details.) The LUCAM_SNAPSHOT structure allows the setting of the following parameters that control the taking of a snapshot and the timing of triggers:

Trigger Mode (useHwTrigger): There are two types of snapshot triggering, hardware and software. When enabled, the snapshot will be triggered when the trigger input signal is detected after a "Take Snapshot" API is called (the API blocks until it times out or until the trigger occurs and the frame of data is returned). When disabled, the API function itself triggers the snapshot and returns the frame of data. The hardware trigger is expected on Pin 9 of the External Interface Header as described above. The software trigger is initiated from within the API "Take Snapshot" functions (for more details see the API documentation.)

Trigger Delay (exposureDelay): A delay in milliseconds from the trigger (hardware or software) to the start of frame exposure can be set from 0 to 10.8 ms, in 42.6 us increments. The minimum delay from the trigger input, to the start of exposure, is 0.8 us, with an error of ± 0.25 us.

Strobe Mode (useStrobe): In concert with either triggering mode, a user may also trigger an external strobe light synchronized to the frame exposure. When this parameter is enabled, the strobe signal pulse will be initiated on Pins 1 and 3 as described above. In this case, a strobe delay should be defined.

Strobe Delay (strobeDelay): A delay in milliseconds from the trigger (hardware or software) to the strobe pulse (rising edge for ACTIVE HIGH, falling edge for ACTIVE LOW) can be set from 0 to 10.8 ms, in 42.6 us increments.

Exposure Time (exposure): The length of time in milliseconds to expose the image before readout begins. The exposure can be set from 0 to 2.3 seconds, in 42.6 us increments.

3.8 External Power

The camera is normally powered via the USB cable, which nominally supplies 5 Volts. A power adapter can also be used to power the camera, in cases where the USB cable does not supply power (e.g. from a Laptop computer or non-powered USB hub.)

The external power adapter must adhere to the following specifications:

1. 6 Volts DC Regulated
2. 1000 mA Minimum Current Rating
3. 2.1 mm tip
4. Center positive (+)

3.9 Lens Mount

The camera is equipped with an industry standard C-Mount lens mount.

4

Application Programming Interface User's Guide

4.1 General Overview

The API is composed of various functions that are used to control the camera, query its state and acquire data from it. There are two groups of functions, namely, a basic group and an advanced group.

The functions in the basic group are very simple to understand and use, and the majority of an application's functionality can be realized quickly using only these functions.

The functions of the advanced group are more powerful and provide greater flexibility and tighter control over the camera's operation; however, they require a more in-depth understanding of the inner workings of the camera. This understanding can be gained from the information in the previous sections. If you have questions that are not covered in this manual, you can e-mail technical support group at:

support@lumenera.com

The following section provides guidance for performing specific and common tasks using the API. The source code provided with the Developer's Kit contains concrete examples of these tasks. For details and further information about calling specific API functions, see the *Lumenera USB Camera API Reference Manual*.

4.2 Basic Tasks

4.2.1 Connecting and Disconnecting

In order to communicate with a camera you must first open a connection to it and obtain its handle. This handle is used as an input parameter to most of the other API functions. The function used for this task is ***LucamCameraOpen***. When you are completely finished with the camera, you should terminate the connection using ***LucamCameraClose***.

Multiple cameras may be attached to the computer at one time. You can obtain the number of cameras attached prior to connecting with any of them using the ***LucamNumCameras*** function. The ***LucamEnumCameras*** function can be used to obtain the unique serial numbers and camera type for all cameras attached to the computer. Thus, it is possible for a specific camera to be opened.

4.2.2 Query the Camera

Several functions exist to obtain information about the camera. To obtain the version information for the camera, its driver and the API, you can use the ***LucamQueryVersion*** function.

To determine the USB connection type (e.g. USB2.0 or USB1.1) you can call ***LucamQueryExternInterface***.

To determine the available frame rates for the camera you can use the ***LucamEnumAvailableFrameRates*** function.

4.2.3 Preview Video

After a camera has been opened, a video preview can be displayed simply by calling ***LucamStreamVideoControl***. This function takes three parameters; 1) the handle to the camera, 2) a control flag (set to START_DISPLAY) and, 3) a window handle (if NULL a window is automatically created for the video display).

More advanced applications may wish to create their own display window. The handle to this window can be passed to the function and the video will be previewed in it.

Pausing or stopping the preview can be achieved using the same function with the control flag set to PAUSE_STREAM or STOP_STREAMING. When the video stream is stopped, the display window is removed (unless you have created your own display window.)

While video is previewing, the displayed frame rate can be obtained using the function ***LucamQueryDisplayFrameRate***.

4.2.4 Adjusting the Video

The API allows for the simple adjustment of several of the video properties by providing the ability to pop up one of two pre-defined dialogs. These dialogs are based on the DirectShow libraries. ***LucamDisplayPropertyPage*** will pop up a dialog for adjusting image properties (exposure, gain, etc.) ***LucamDisplayVideoFormatPage*** will pop up a dialog for adjusting the video format (resolution, pixel bit depth, etc.)

4.2.5 Configuring Video Format

At any time when the camera is not streaming video, you can configure the video format using the ***LucamSetFormat*** function. This will allow you to select the following video properties:

1. Subwindow size
2. Subwindow position
3. Subsampling
4. Pixel Format
5. Video Frame Rate

There are some constraints and limitations on and associated with the above properties.

1. The subwindow size is provided as the window width and height. Both the width and height must be a multiple of 8.
2. The subwindow position is provided as the x and y coordinates of the top left corner of the subwindow. Both x and y must be multiples of 8.
3. Subsampling must be the same for both x and y directions and is limited to 2x2 or 4x4.
4. The Pixel Format is provided as LUCAM_PF_8 or LUCAM_PF_16 and indicates the bit depth of the data. Using LUCAM_PF_16 doubles the amount of data coming from the camera and will cause the maximum frame rate to be cut in half. Although there are 16 bits per pixel in this mode, only 10, 12, or 14 bits of data are valid (depending on the maximum bit depth of the particular camera being used.)
5. The video frame rate provided may not be exactly as requested. The closest smaller available rate will be provided.

4.2.6 Grab Video Data

As long as video streaming is turned on (with the ***LucamStreamVideoControl*** function), video data can be grabbed from the camera. The function to use for this is ***LucamTakeVideo***. The number of frames must be specified as well as a pointer to a buffer that will accept the data. The data returned from the camera is in its raw form with each byte (for LUCAM_PF_8) or word (for LUCAM_PF_16) returned from the camera representing a single pixel in the image. For color cameras, the data is still in the Bayer format as described in a previous section.

It's not necessary to be previewing the data to capture it. Grabbing video while the preview is turned off is much more efficient and leaves the CPU free to do other processing tasks.

4.2.7 Take a Snapshot (or many)

Single snapshots can be taken using one of several functions. The ***LucamTakeSnapshot*** function can be used any time a camera is open to asynchronously take a single snapshot using the camera's snapshot mode. Even if the camera is streaming video, the function will handle the stream, stopping and then restarting it as necessary. The overhead associated with managing the stream means this function is not extremely fast. However, it is a very simple way to obtain snapshots when time is not critical.

For quicker snapshot capturing, individual functions exist to enable the snapshot mode (***LucamEnableFastFrame***), take snapshots (***LucamTakeFastFrames***) and then disable the snapshot mode (***LucamDisableFastFrames***) separately. The use of these functions helps eliminate overhead associated with managing the streaming video mode. When the Enable function is called, streaming is stopped, and then snapshots can be captured repeatedly at a faster rate than with ***LucamTakeSnapshot***.

As with the "TakeVideo" functions, the data returned from the camera is in its raw, unprocessed form.

4.2.8 Processing Images

For monochrome cameras, the data that arrives from the camera is ready for user processing, display or capture to disk.

Data from color cameras, on the other hand, is in the raw Bayer format and will typically need to be converted to 24-bit RGB before being user processed, displayed or captured to disk. This conversion can be accomplished using the ***LucamConvertFrameToRgb24*** function.

The two additional functions ***LucamConvertFrameToRgb32*** and ***LucamConvertFrameToRgb48*** are available to convert the raw Bayer data into 32 and 48-bit RGB data respectively.

The ***LucamConvertBmp24ToRgb24*** function will convert the byte order for each pixel from .bmp file standard Blue, Green, Red to Red, Green, Blue.

The color data can also be converted to 8-bit and 16-bit monochrome (Greyscale) using the ***LucamConvertFrameToGreyscale8*** and ***LucamConvertFrameToGreyscale16*** respectively.

4.2.9 Save Image to Disk

Saving images to disk can be accomplished with the ***LucamSaveImage*** or ***LucamSaveImageW*** functions. Once a frame of data has been taken from the camera (video or snapshot) it can be saved in one of several image formats. (For color cameras, first convert to 24-bit RGB to obtain a proper color image.)

The available formats are:

1. Raw Data (.raw) - no header, no compression

2. Bitmap (.bmp) – Windows bitmap file
3. TIFF (.tif) – Tagged Image File Format
4. JPEG (.jpg) – JPEG compression
5. PNG (.png) – Portable Network Graphics Format

4.2.10 Setting and Getting Camera Properties

Camera properties are items such as exposure, gain, contrast, etc. There are only three functions associated with camera properties. ***LucamSetProperty*** is used to set the value of a camera property. ***LucamGetProperty*** is used to get the value of a camera property. ***LucamPropertyRange*** will return the valid range and default value for a given camera property. It is important to note that not all properties are available for every camera supported by the API. The table below indicates which properties are available for each camera model.

Table 6 - Property Availability By Camera Model

Property	LU-050	LU-070	LU-080	LU-100	LU-110	LU-120	LU-130
Brightness	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Contrast	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Hue	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Saturation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sharpness	No	No	No	No	No	No	No
Gamma	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Exposure	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gain	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Red Gain	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Blue Gain	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Green1 Gain	Yes	Yes	Yes	No	Yes	Yes	Yes
Green2 Gain	Yes	Yes	Yes	No	Yes	Yes	Yes

Property	LU-160	LU-170	LU-200	LU-270	LU-330	LU-370	LU-620
Brightness	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Contrast	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Hue	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Saturation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sharpness	No	No	No	No	No	No	No
Gamma	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Exposure	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gain	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Red Gain	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Blue Gain	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Green1 Gain	Yes	Yes	No	Yes	Yes	Yes	Yes

Green2 Gain	Yes	Yes	No	Yes	Yes	Yes	Yes

4.3 Advanced Tasks

4.3.1 Manage Your Own Video Display Window

When the ***LucamStreamVideoControl*** is used to start previewing video data, the window is created automatically. Using ***LucamCreateDisplayWindow*** you can create your own display window and specify its size and location on the screen.

4.3.2 Custom Color Correction Matrix

For color cameras, the data returned from the camera is in the Bayer format. However, this data needs to be color corrected to obtain true color. Normally, one of the standard matrices would be selected from the available ones in the API, but there may be instances where a custom matrix is desired. For example, a color image can be converted to monochrome with the appropriate matrix. Changing the Hue and Saturation can also be performed using a custom matrix. In order to apply a custom matrix, it must first be defined. This is done using the ***LucamSetupCustomMatrix*** function. To make use of this custom matrix in video mode, you set the LUCAM_PROP_CORRECTION_MATRIX property to LUCAM_CM_CUSTOM using ***LucamSetProperty***. If you want the custom matrix to be used when converting raw images from the camera to RGB24 using ***LucamConvertFrameToRgb24***, set the CorrectionMatrix field of the LUCAM_CONVERSION structure to LUCAM_CM_CUSTOM.

4.3.3 Custom Look-Up-Tables

The camera has a built-in 8-bit LUT, which is used by the Brightness, Contrast and Gamma properties. You can provide your own customized LUT, which will be applied to all data coming from the camera, using the ***LucamSetup8bitsLUT*** function. If you provide your own LUT, the Brightness, Contrast and Gamma properties should not be used; otherwise they will overwrite the custom LUT.

4.3.4 Video Callback functions

The Video Callback feature allows you to supply your own function that will be called for every frame of data that arrives from the camera, giving you access to the data stream for frame-by-frame processing. This can be used to provide a graphic overlay, for example, or false color to a monochrome image etc.

There are two functions that can be used to register your callback. The first is named ***LucamAddStreamingCallback***. The second is named ***LucamAddRgbPreviewCallback***. The only difference between them is the data format of the frame of data that is passed to them. The first function will be passed the raw data as it comes out of the camera. The second function will be passed the data after it has been processed into RGB data. (For monochrome cameras, there will be no difference between the data passed for either function.)

The ***LucamQueryRgbPreviewPixelFormat*** function can be used to determine the bit depth of the RGB data (either 24 bpp or 32 bpp).

The callback function can be removed using either ***LucamRemoveStreamingCallback*** or ***LucamRemoveRgbPreviewCallback*** depending on which function was used to add it.

4.3.5 MultipleCamera, Simultaneous Image Capture

Several cameras can be connected to the same computer and used to capture an image at the same time. This is accomplished using a set of three API functions. ***LucamEnableSynchronousSnapshots*** enables this mode for all the cameras. ***LucamTakeSynchronousSnapshots*** is used to perform the snapshot capture. ***LucamDisableSynchronousSnapshots*** is used to disable this mode for all the cameras.

4.3.6 Non-Volatile User Accessible Camera Memory

The camera contains a 1024-byte area of non-volatile memory that is user-accessible. The ***LucamPermanentBufferWrite*** function can be used to write arbitrary data to the memory area that won't get erased when the camera is powered down. The ***LucamPermanentBufferRead*** function can be used to read the memory area. There is a limit of 100,000 times that the memory area can be overwritten reliably.