POINT GREY
RESEARCH

ActiveFlyCap

ActiveX Control for FlyCapture Reference

Version 1.2
Revised May 15, 2008

Point Grey Research Inc.
8866 Hudson Street • Vancouver, BC • Canada • V6P 4N2 • **T** (604) 730-9937 • **www.ptgrey.com**

Software Warranty

Point Grey Research warrants to the Original Purchaser, for a period of one (1) year from date of purchase that:

1. The diskette on which the Software is furnished and the accompanying documentation are not defective;
2. The Software is properly recorded upon the diskettes enclosed;
3. The documentation is substantially complete and contains all the information Point Grey Research deems necessary to use the Software;
4. The Software functions substantially as described in the documentation.

Point Grey Research, Inc.'s entire liability and the Original Purchaser's exclusive remedy shall be the replacement of any diskette or documentation not meeting these warranties. On such an occasion, a copy of the paid receipt accompanied with the faulty diskette or documentation must be returned to Point Grey Research, Inc. or an authorized dealer.

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

Trademarks

Point Grey Research, PGR, the Point Grey Research, Inc. logo, Digiclops, Dragonfly, Dragonfly Express, Firefly, Flea, Grasshopper, Ladybug, Scorpion and Triclops are trademarks or registered trademarks of Point Grey Research, Inc. in Canada and other countries.

Table of Contents

1 Overview

The ActiveFlyCap control allows users to easily insert an ActiveX control into a GUI application and control Point Grey cameras easily without having to worry about writing additional code to draw images onto the screen.

1.1 Supported Programming Languages

The following programming languages are supported:

- Visual Basic 6
- Visual Basic.Net
- Visual C#
- Visual C++

Any language capable of interacting with ActiveX controls (e.g. in a Windows Form) should be able to interface with ActiveFlyCap.

1.2 Manual Registration

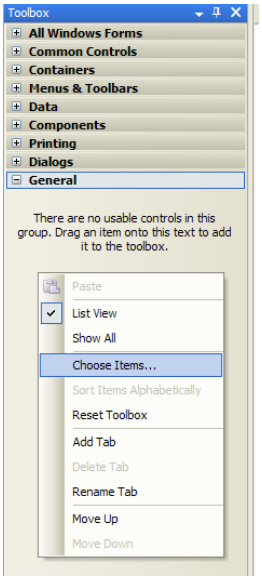
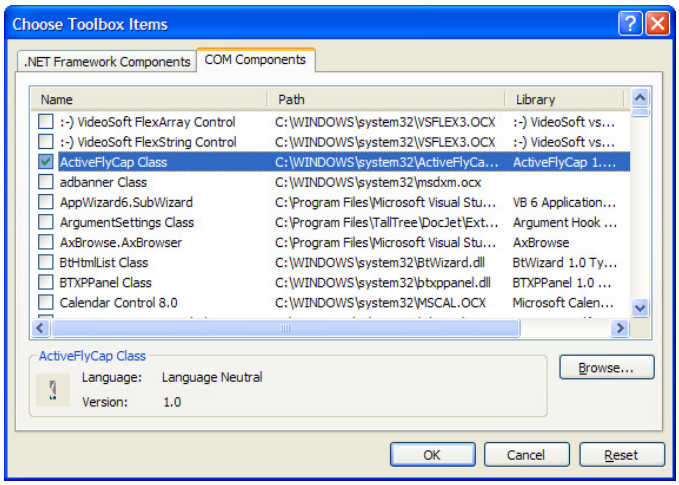
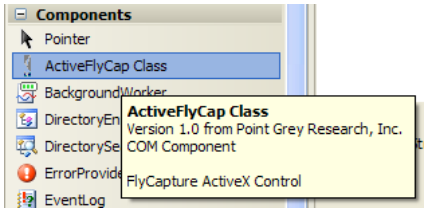
In the event that the control must be manually registered, call regsvr32 with the name of the control (`ActiveFlyCap.dll`). The FlyCapture DLLs (`PGRFlyCapture.dll` and `PGRFlyCaptureGUI.dll`) should also be present in the system.

1.3 Supporting downloads

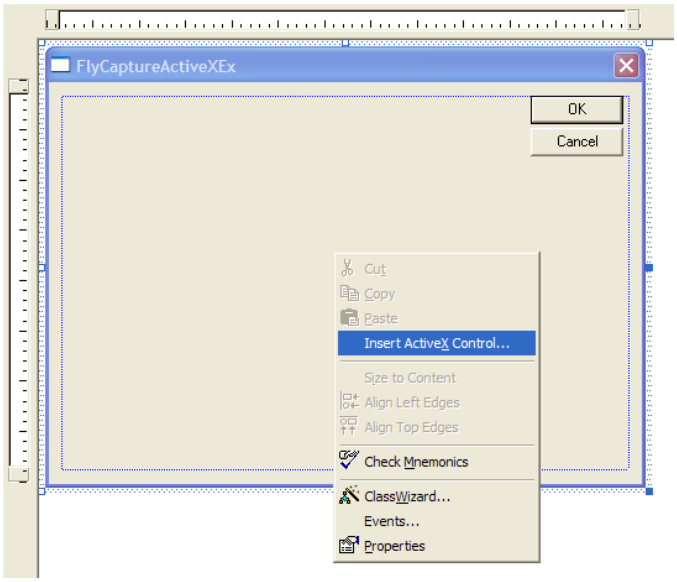
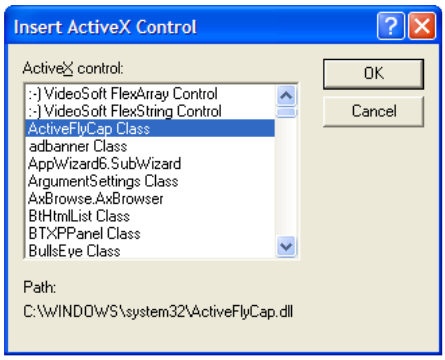
It may be necessary to install the Microsoft Visual C++ 2005 SP1 Redistributable Package. The package can downloaded directly from Microsoft from the following URL: http://www.microsoft.com/downloads/details.aspx?FamilyID=200b2fd9-ae1a-4a14-984d-389c36f85647&displaylang=en.

1.4 Adding the Control

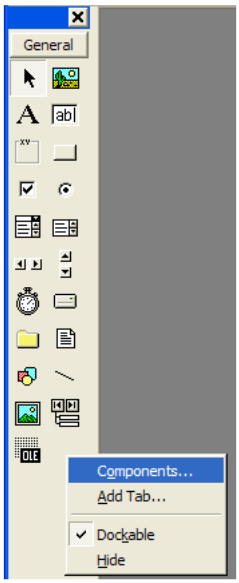
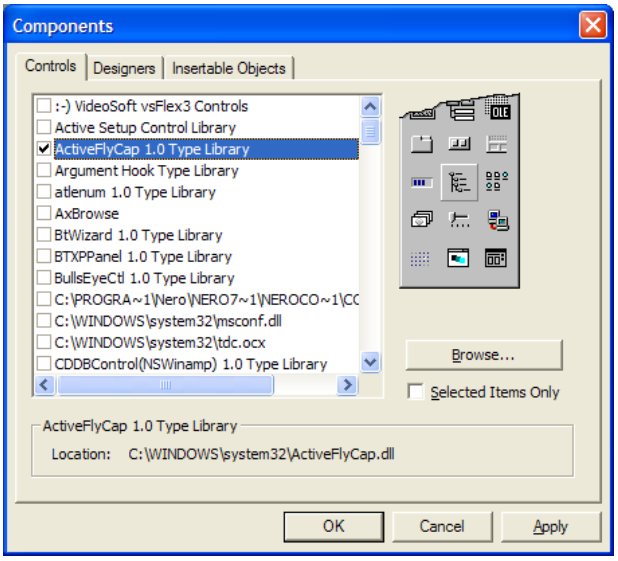
1.4.1 Visual Studio 2005

Step	Instruction	Screenshot
1	In a GUI-based application, right click the *Toolbox* and select *Choose Items.*	
2	Under *COM Components*, make sure that the ActiveFlyCap class is selected.	
3	The ActiveFlyCap control component will now appear in the *Toolbox.*	

1.4.2 Visual C++ 6

Step	Instruction	Screenshot
1	Right click a dialog in the dialog editor and select *Insert ActiveX Control.*	
2	Select the ActiveFlyCap class and click *Ok.*	

1.4.3 Visual Basic 6

Step	Instruction	Screenshot
1	Right click the *Toolbox* and select *Components.*	
2	Select the ActiveFlyCap Type Library and click *Ok.*	
3	The ActiveFlyCap component will now appear in the *Toolbox.*	

1.5 Grab Modes

There are 3 grab modes available for use in ActiveFlyCap.

1.5.1 Free Running

This mode simply starts the camera and automatically draws images to the screen. This mode is ideal if accessing the image data directly is not required.

1.5.2 LockLatest

This mode locks the latest image that hasn't been seen by the user. This means that there may be a possibility that images may be lost if the PC is unable to perform processing quickly enough.

1.5.3 LockNext

This mode locks the oldest image that hasn't been seen by the user. Provided that the PC is fast enough to process images faster than the frame rate of the camera, this mode guarantees that no images are lost.

2 Declarations

2.1 Structures

In order to pass multiple arguments to the control, structures are used to contain data that will be passed between the application, the ActiveFlyCap control and the FlyCapture library. Structures do not have to be populated before use. Any incoming parameters must be supplied as a separate argument when the function is called.

2.1.1 CameraInfo

Type	Name	Description
LONG	lSerialNumber	Serial number of the camera
CameraType	camType	Camera type (B&W or Color)
CameraModel	camModel	Model of camera
BSTR	bstrModelName	Model name
BSTR	bstrVendorName	Vendor name
BSTR	bstrSensorInfo	Sensor info
LONG	lDCAMVer	DCAM version. Divide this number by 100 for the actual number
LONG	lNodeNum	Node number
LONG	lBusNum	Bus number
BusSpeed	maxBusSpeed	Maximum bus speed

2.1.2 ImageInfo

Type	Name	Description
LONG	lRows	Number of rows in the image
LONG	lCols	Number of columns in the image
LONG	lRowInc	Number of bytes per row in the image
LONG	lNumImages	Number of images contained within
DCAMVideoMode	videoMode	Video mode
PixelFormat	pixelFmt	Pixel format

2.1.3 AbsPropertyStruct

Type	Name	Description
BOOL	bOnePush	One push status
BOOL	bOnOff	On/off status
BOOL	bAuto	Auto status
FLOAT	fValue	Value

2.1.4 AbsPropertyRangeStruct

Type	Name	Description
BOOL	bSupported	Whether the property has absolute value support
FLOAT	fMin	Minimum value

FLOAT	fMax	Maximum value
BSTR	bstrUnits	Units for the property
BSTR	bstrUnitAbbv	Abbreviated units for the property

2.1.5 PropertyStruct

Type	Name	Description
BOOL	bOnePush	One push status
BOOL	bOnOff	On/off status
BOOL	bAuto	Auto status
LONG	lValueA	Value A
LONG	lValueB	Value B

2.1.6 PropertyRangeStruct

Type	Name	Description
BOOL	bSupported	Whether the property has absolute value support
BOOL	bOnePush	One push support
BOOL	bReadOut	Whether the value can be read out
BOOL	bOnOff	On/off support
BOOL	bAuto	Auto support
BOOL	bManual	Manual support
FLOAT	fMin	Minimum value
FLOAT	fMax	Maximum value

2.1.7 TriggerStruct

Type	Name	Description
BOOL	bOnOff	On/off support
LONG	lPolarity	Polarity
LONG	lSource	Source
LONG	lRawValue	Raw value
LONG	lMode	Trigger mode
LONG	lParameter	Parameter

2.1.8 StrobeStruct

Type	Name	Description
BOOL	bOnOff	On/off support
BOOL	bPolarity	Polarity
LONG	lDelay	Delay
LONG	lDuration	Duration

2.1.9 Format7QueryStruct

Type	Name	Description
BOOL	bAvailable	Whether the mode is available in Format 7
LONG	lMaxImagePixelsWidth	Maximum horizontal width
LONG	lMaxImagePixelsHeight	Maximum horizontal height
LONG	lPixelUnitHorz	Horizontal step size of the Format 7 image
LONG	lPixelUnitVert	Vertical step size of the Format 7 image
LONG	lOffsetUnitHorz	Horizontal step size of offset of the Format

		7 image
LONG	lOffsetUnitVert	Vertical step size of the offset of the Format 7 image
LONG	lPixelFormats	Bit field containing supported pixel formats

2.1.10 Format7SettingsStruct

Type	Name	Description
LONG	lMode	Format 7 mode
LONG	lImagePosLeft	Left offset
LONG	lImagePosTop	Top offset
LONG	lImageWidth	Image width
LONG	lImageHeight	Image height
LONG	lPacketSizeBytes	Packet size in bytes
FLOAT	fBandwidth	Percentage of bandwidth

2.1.11 PacketInfo

Type	Name	Description
LONG	lMinSizeBytes	Minimum packet size in bytes
LONG	lMaxSizeBytes	Maximum packet size in bytes

3 Properties

The property get/set functions always return a HRESULT in C++. The HRESULT can have several possible values:

Name	Description
S_OK	Success
E_FAIL	Failure
E_INVALIDARG	Invalid argument
E_POINTER	Invalid pointer

In VB and C#, the return value is the value of the property itself. An error or exception handler should always be used appropriately.

3.1 Control Properties

3.1.1 EnableRightClickMenu

Enables or disables the right click menu. The right click menu contains basic functionality such as starting and stopping the camera, as well as showing the camera control dialog. It is set to 0 (off) by default.

3.1.2 Display

Draw all incoming images onto control.

3.1.3 AutoResize

Automatically resize the image to fit the control. If the value is FALSE, then scroll bars will appear to enable movement around the image.

3.1.4 Font

The font to be used when drawing text to the screen.

3.2 General Camera Properties

3.2.1 Camera

Get or set the index of the currently selected camera. This **does not** start the camera.
-1 is returned if there is no camera currently selected.

3.2.2 Start

Start grabbing images. This will enable isochronous data transmission from the camera to the PC. Depending on the grab mode, this may or may not automatically grab images. This **does not** draw the image to screen.
Setting this to 0 will stop the camera, ending isochronous data transmission.

3.2.3 VideoMode

Get or set the current video mode.
Setting the video mode is only possible if the Start property is 0.

3.2.4 FrameRate

Get or set the current frame rate.
Setting the frame rate is only possible if the Start property is 0.

3.2.5 AsyncBusSpeed

Get or set the current asynchronous bus speed (`S100`, `S200`, `S400`, `S800`).
Setting the bus speed is only possible if the Start property is 0.

3.2.6 IsochBusSpeed

Get or set the current isochronous bus speed (`S100`, `S200`, `S400`, `S800`).
Setting the bus speed is only possible if the Start property is 0.

3.2.7 GrabTimeout

Get or set the grab timeout.
0 is a non-blocking grab call.
-1 is equivalent to `FLYCAPTURE_INFINITE`.

4 Methods

4.1 Control Methods

4.1.1 DrawSingleImage

Argument	Type	Description
bConvert	[in] BOOL	Perform color processing on raw image

Draw the latest image to the control. This is only used when the Display property is set to 0.
If the internal raw image has been modified after using GetImageData() or GetImagePtr(), then the bConvert argument should be set to TRUE to perform color processing on the raw image again.

4.2 GUI Methods

4.2.1 ToggleSettingsWindowState

Argument	Type	Description
N/A	N/A	N/A

Toggle the display of the camera control dialog.

4.2.2 GetSettingsWindowState

Argument	Type	Description
bShowing	[out, retval] BOOL*	Whether the camera control dialog is currently being shown

Get the status of the camera control dialog.

4.2.3 ShowCameraSelectionModal

Argument	Type	Description
bDialogStatus	[out, retval] BOOL*	Whether the Ok or Cancel button was clicked. TRUE for Ok, FALSE for Cancel

Show the camera selection dialog. If the camera associated with this control is grabbing images, it will be stopped. Selecting a camera from the dialog will set the active camera to the selected one.

4.3 Camera Methods

4.3.1 GetCameraInfo

Argument	Type	Description
pInfo	[out, retval] CameraInfo*	A CameraInfo structure containing the camera information

Get camera info.

4.3.2 CheckVideoMode

Argument	Type	Description
videoMode	[in] DCAMVideoMode	The DCAM video mode to test against
frameRate	[in] DCAMFrameRate	The DCAM frame rate to test against
pbSupported	[out, retval] BOOL*	Whether the video mode is supported by the camera

Check if the specified video mode and frame rate combination is supported.

4.3.3 GetFormat7PacketInfo

Argument	Type	Description
lMode	[in] LONG	The Format 7 mode to check
lWidth	[in] LONG	The width of the Format 7 image
lHeight	[in] LONG	The height of the Format 7 image
pixelFormat	[in] PixelFormat	The pixel format of the Format 7 image
pPacketInfo	[out, retval] PixelFormat*	A PacketInfo structure containing the packet information

Get the Format 7 packet info for a particular mode, image size and pixel format.

4.3.4 GetPacketInfo

Argument	Type	Description
videoMode	[in] DCAMVideoMode	The DCAM video mode to test against
frameRate	[in] DCAMFrameRate	The DCAM frame rate to test against
pPacketInfo	[out, retval] PixelFormat*	A PacketInfo structure containing the packet information

Get the packet info for a particular DCAM mode

4.3.5 GetImageInformation

Argument	Type	Description
pImageInfo	[out, retval] ImageInfo*	A ImageInfo structure containing the image information

Get image information about the current image.

4.3.6 GetImageData

Argument	Type	Description
type	[in] ImageType	The type of image to return (raw or color processed)
pArray	[out, retval] VARIANT*	An array containing the image data

Get the image data for the current image. This should not be used in .NET languages as a copy of the data will be automatically made instead.

4.3.7 GetImagePtr

Argument	Type	Description
type	[in] ImageType	The type of image to return (raw or color processed)
pValue	[out, retval] VARIANT*	An pointer to the array containing the image data

Get a pointer to the image data. Although the value returned is a LONG, it can be cast to a pointer to access the image data directly.

4.4 1394 Bus Methods

4.4.1 BusCameraCount

Argument	Type	Description
plNumCameras	[out, retval] LONG*	The number of cameras detected

Get the number of cameras on the bus.

4.4.2 BusEnumerateCameras

Argument	Type	Description
pList	[out, retval] VARIANT*	An array containing the model names and serial numbers of the cameras on the bus

Get an array containing a list of cameras on the bus. The contents of the array are strings containing the camera name as well as the serial number in parentheses.

4.5 Grab Methods

4.5.1 SetGrabMode

Argument	Type	Description
mode	[in] GrabMode	The grab mode that the control is to be set to

Set the grab mode from the following:

- FreeRunning
- LockLatest
- LockNext

The default grab mode is FreeRunning.

4.5.2 GetGrabMode

Argument	Type	Description
pMode	[out, retval] GrabMode*	The current grab mode

Get the current grab mode.

4.5.3 GrabImage

Argument	Type	Description
pError	[out, retval] GrabError*	The error returned from the grab

Grab the next image from the camera. The image that is returned depends on the grab mode that is set. If LockLatest mode is set, then the newest image will be captured. If LockNext is set, then the oldest image will be captured. This call is not valid if the camera is in free running mode.

4.6 Image Saving Methods

4.6.1 SaveImage

Argument	Type	Description
bstrFilename	[in] BSTR	The path to save the file to
fileFormat	[in] ImageFileFormat	The file format to save the image in

Save the current image.

4.6.2 SetJPEGCompressionQuality

Argument	Type	Description
lQuality	[in] LONG	JPEG compression quality (0-100)

Set the JPEG compression quality.

4.7 Format 7 Methods

4.7.1 QueryFormat7

Argument	Type	Description
lMode	[in] LONG	The Format 7 mode to retrieve information from

pStruct	[out, retval] Format7QueryStruct*	A Format7QueryStruct containing Format 7 information for the specified mode

Get the possible settings for a particular Format 7 mode.

4.7.2 GetFormat7Settings

Argument	Type	Description
pSettingsStruct	[out, retval] Format7SettingsStruct*	A Format7SettingsStruct containing the current Format 7 settings

Get the current Format 7 settings.

4.7.3 SetFormat7Settings

Argument	Type	Description
pSettingsStruct	[in] Format7SettingsStruct*	A Format7SettingsStruct containing the Format 7 settings to be set

Set the current Format 7 settings. If the image offsets are the only parameter being changed, then this can be done when the camera is started. Otherwise, the camera must be stopped in order for the call to succeed.

4.8 Camera Property Methods

These methods enable the user to get the camera properties.

4.8.1 GetCameraAbsProperty

Argument	Type	Description
camProp	[in] CameraProperty	The camera property to get
pStruct	[out, retval] AbsPropertyStruct*	A AbsPropertyStruct containing the data about the specified property

Get detailed absolute property.

4.8.2 SetCameraAbsProperty

Argument	Type	Description
camProp	[in] CameraProperty	The camera property to set
pStruct	[in] AbsPropertyStruct*	A AbsPropertyStruct containing the data to be set for the specified property

Set detailed absolute property.

4.8.3 GetCameraAbsPropertyRange

Argument	Type	Description
camProp	[in] CameraProperty	The camera property to get
pStruct	[out, retval]	A AbsPropertyRangeStruct containing the

	AbsPropertyRangeStruct*	data about the specified property

Get absolute property range.

4.8.4 GetCameraProperty

Argument	Type	Description
camProp	[in] CameraProperty	The camera property to set
pStruct	[out,retval] PropertyStruct*	A PropertyStruct containing the data to be set for the specified property

Get detailed property.

4.8.5 SetCameraProperty

Argument	Type	Description
camProp	[in] CameraProperty	The camera property to set
pStruct	[in] PropertyStruct*	A PropertyStruct containing the data to be set for the specified property

Set detailed property.

4.8.6 GetCameraPropertyRange

Argument	Type	Description
camProp	[in] CameraProperty	The camera property to get
pStruct	[out, retval] PropertyRangeStruct*	A PropertyRangeStruct containing the data about the specified property

Get detailed property range.

4.9 Trigger / Strobe Methods

4.9.1 GetTrigger

Argument	Type	Description
pStruct	[out, retval] TriggerStruct*	A TriggerStruct containing the trigger data

Get trigger status.

4.9.2 SetTrigger

Argument	Type	Description
pStruct	[in] TriggerStruct*	A TriggerStruct containing the trigger data to be set

Set trigger status.

4.9.3 SetTriggerBroadcast

Argument	Type	Description
pStruct	[in] TriggerStruct*	A TriggerStruct containing the trigger data to be set

Set and broadcast trigger status.

4.9.4 GetStrobe

Argument	Type	Description
lMode	[in] LONG	The strobe source to retrieve
pStruct	[out, retval] StrobeStruct*	A StrobeStruct containing the strobe data

Get strobe status.

4.9.5 SetStrobe

Argument	Type	Description
lMode	[in] LONG	The strobe source to set
pStruct	[in] StrobeStruct*	A StrobeStruct containing the strobe data to be set

Set strobe status.

4.9.6 SetStrobeBroadcast

Argument	Type	Description
lMode	[in] LONG	The strobe source to set
pStruct	[in] StrobeStruct*	A StrobeStruct containing the strobe data to be set

Set and broadcast strobe broadcast.

4.10 Memory Channel Methods

4.10.1 GetNumMemoryChannels

Argument	Type	Description
plNumChannels	[out, retval] LONG*	The number of available memory channels

Get the number of memory channels available.

4.10.2 GetCurrentMemoryChannel

Argument	Type	Description
plCurrentChannel	[out, retval] LONG*	The current memory channel

Get the number of the memory channel currently being used.

4.10.3 SaveToMemoryChannel

Argument	Type	Description
lCurrentChannel	[in] LONG*	The memory channel to save to

Save current settings to the specified memory channel.

4.10.4 RestoreFromMemoryChannel

Argument	Type	Description
lChannel	[in] LONG*	The memory channel to load from

Restore settings from the specified memory channel.

4.11 Register Methods

4.11.1 GetCameraRegister

Argument	Type	Description
lRegister	[in] LONG	The register to retrieve
plValue	[in] LONG*	The value of the register

Get the register value at the specified offset.

4.11.2 SetCameraRegister

Argument	Type	Description
lRegister	[in] LONG	The register to retrieve to set
lValue	[in] LONG	The value of the register

Set the register value at the specified offset.

4.11.3 ReadRegisterBlock

Argument	Type	Description
sAddrHigh	[in] LONG	The top 16 bits of the 48-bit absolute address to read
lAddrLow	[in] LONG	The bottom 32 bits of the 48-bit absolute address to read
pBuffer	[in] LONG*	The buffer to receive the data
lBufferLength	[in] LONG	The length of the data buffer, in quadlets

Get the register block at the specified offset.

4.11.4 WriteRegisterBlock

Argument	Type	Description

sAddrHigh	[in] LONG	The top 16 bits of the 48-bit absolute address to write
lAddrLow	[in] LONG	The bottom 32 bits of the 48-bit absolute address to write
pBuffer	[in] LONG*	The buffer of data to be written
lBufferLength	[in] LONG	The length of the data buffer, in quadlets

Set the register block at the specified offset.

4.12 Drawing Methods

All coordinates used for drawing are relative to the image origin before any auto-resizing (if applicable) has taken place.

4.12.1 DrawEllipse

Argument	Type	Description
X1	[in] SHORT	X coordinate of the start point
Y1	[in] SHORT	Y coordinate of the start point
X2	[in] SHORT	X coordinate of the end point
Y2	[in] SHORT	Y coordinate of the end point
sWidth	[in] SHORT	Width of the line
redVal	[in] SHORT	Red value
greenVal	[in] SHORT	Green value
blueVal	[in] SHORT	Blue value

Draw an ellipse bounded by the coordinates (X1, Y1) and (X2, Y2).

4.12.2 DrawLine

Argument	Type	Description
X1	[in] SHORT	X coordinate of the start point
Y1	[in] SHORT	Y coordinate of the start point
X2	[in] SHORT	X coordinate of the end point
Y2	[in] SHORT	Y coordinate of the end point
sWidth	[in] SHORT	Width of the line
redVal	[in] SHORT	Red value
greenVal	[in] SHORT	Green value
blueVal	[in] SHORT	Blue value

Draw a line from the coordinates (X1, Y1) to (X2, Y2).

4.12.3 DrawPixel

Argument	Type	Description
XPos	[in] SHORT	X coordinate of the pixel
YPos	[in] SHORT	Y coordinate of the pixel
redVal	[in] SHORT	Red value
greenVal	[in] SHORT	Green value
blueVal	[in] SHORT	Blue value

Draw a pixel at the coordinates (XPos, YPos) with the specified RGB value. This does not overwrite the actual image data.

4.12.4 DrawRectangle

Argument	Type	Description
X1	[in] SHORT	X coordinate of the start point
Y1	[in] SHORT	Y coordinate of the start point
X2	[in] SHORT	X coordinate of the end point
Y2	[in] SHORT	Y coordinate of the end point
sWidth	[in] SHORT	Width of the line
redVal	[in] SHORT	Red value
greenVal	[in] SHORT	Green value
blueVal	[in] SHORT	Blue value

Draw a rectangle bounded by the coordinates (X1, Y1) and (X2, Y2).

4.12.5 DrawText

Argument	Type	Description
XPos	[in] SHORT	X coordinate of the start point
YPos	[in] SHORT	Y coordinate of the start point
bstrText	[in] BSTR	The text to be written
redVal	[in] SHORT	Red value
greenVal	[in] SHORT	Green value
blueVal	[in] SHORT	Blue value

Draw the specified text at the coordinates (XPos, YPos).

5 Events

5.1 General Events

5.1.1 CameraArrival

A new camera has arrived on the bus. Depending on the bus topology, the index used to refer to cameras may have changed. It is highly recommended that BusEnumerateCameras() be called to refresh the camera list.

The argument received is the serial number of the camera that has arrived.

5.1.2 CameraRemoval

A camera has been removed from the bus. Depending on the bus topology, the index used to refer to cameras may have changed. It is highly recommended that BusEnumerateCameras() be called to refresh the camera list.

The argument received is the serial number of the camera that has been removed.

5.1.3 BusReset

A bus reset has occurred.

5.1.4 Image

A new image has been received from the camera. GetImageInformation() can be called when this message is received in order to obtain more information about the image.

6 Additional Resources

For more detailed information regarding advanced camera functionality such as trigger modes, please see the documentation included in your FlyCapture install. These files can be typically be found in `C:\Program Files\Point Grey Research\PGR FlyCapture\doc`.

6.1 Getting Started Manual

A *Getting Started Manual* provides a broad overview of the general capabilities of a specific camera. It contains basic information such as physical dimensions and general camera features, as well as instructions on installation of the camera.

6.2 Technical Reference Manual

A *Technical Reference Manual* provides in-depth information regarding a specific camera. It contains full specifications for a camera, including advanced camera-specific features such as frame buffers, HDR modes and trigger modes.

6.3 Point Grey Digital Camera Register Reference

The *Point Grey Digital Camera Register Reference* provides detailed information about the various features, formats and control parameters implemented by each PGR IEEE-1394 camera. It should be used in conjunction with the camera specific *Technical Reference Manual* or *Getting Started Manual* for a full understanding of a specific camera system.

7 Technical Support Resources

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

Creating a Customer Login Account

The first step in accessing our technical support resources is to obtain a Customer Login Account. This requires a valid name, e-mail address, and camera serial number. To apply for a Customer Login Account go to www.ptgrey.com/support/downloads/.

Knowledge Base

Our on-line knowledge base at www.ptgrey.com/support/kb/ contains answers to some of the most common support questions. It is constantly updated, expanded, and refined to ensure that our customers have access to the latest information.

Product Downloads

Customers with a Customer Login Account can access the latest software and firmware for their cameras from our downloads site at www.ptgrey.com/support/downloads. We encourage our customers to keep their software and firmware up-to-date by downloading and installing the latest versions.

Contacting Technical Support

Before contacting Technical Support, have you:

1. *Read the product documentation and user manual?*
2. *Searched the Knowledge Base?*
3. *Downloaded and installed the latest version of software and/or firmware?*

If you have done all the above and still can't find an answer to your question, contact our Technical Support team at www.ptgrey.com/support/contact/.

8 Contacting Point Grey Research Inc.

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

Email: For all general questions about Point Grey Research please contact us at info@ptgrey.com.

For technical support (existing customers only) contact us at http://www.ptgrey.com/support/contact/.

Knowledge Base: Find answers to commonly asked questions in our knowledge base at http://www.ptgrey.com/support/kb/.

Downloads: Users can download the latest manuals and software from http://www.ptgrey.com/support/downloads/

Main Office:

Mailing Address:
Point Grey Research, Inc.
8866 Hudson Street
Vancouver, BC, Canada
V6P 4N2

Tel: +1 (604) 730-9937
Fax: +1 (604) 732-8231
sales@ptgrey.com

Distributors

USA

Mailing Address:
13749 E. Charter Oak Drive
Scottsdale, AZ USA
85259-2322

Tel: +1 (480) 391-2125
Fax: +1 (480) 391-2125
na-sales@ptgrey.com

Europe

Mailing Address:
Gerstäcker Str. 60
D-81827 München
Germany

Tel: +49 (89) 45463224
Fax: +49 (89) 45463225
eu-sales@ptgrey.com

Japan — ViewPLUS Inc. (http://www.viewplus.co.jp/)

Korea — Cylod Co. Ltd. (http://www.cylod.com)