

pco.labview

LabVIEW driver for pco.camera

This document describes the LabVIEW interface to the pco.camera series.

Copyright © 2005-2015 **pco** AG (called **pco** in the following text), Kelheim, Germany. All rights reserved. **pco** assumes no responsibility for errors or omissions in these materials. These materials are provided "as is" without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. **pco** further does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. **pco** shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. The information is subject to change without notice and does not represent a commitment on the part of **pco** in the future. **pco** hereby authorizes you to copy documents for non-commercial use within your organization only. In consideration of this authorization, you agree that any copy of these documents that you make shall retain all copyright and other proprietary notices contained herein. Each individual document published by **pco** may contain other proprietary notices and copyright information relating to that individual document. Nothing contained herein shall be construed as conferring by implication or otherwise any license or right under any patent or trademark of **pco** or any third party. Except as expressly provided above nothing contained herein shall be construed as conferring any license or right under any **pco** copyright. Note that any product, process, or technology in this document may be the subject of other intellectual property rights reserved by **pco**, and may not be licensed hereunder.

Contents

LABVIEW DRIVER FOR PCO.CAMERA.....	2
1. GENERAL	13
1.1. Conventions.....	13
1.2. Hardware elements, interface connectors	14
2. OVERVIEW AND FUNCTION DESCRIPTION.....	17
2.1. GeneralControlStatus.llb – Camera control	19
2.2. Sensor.llb - Image sensor specifics	20
2.3. TimingControl.llb – Image timing.....	20
2.4. Storage.llb – Camera memory management	22
2.5. RecordingControl.llb – Image recording controls.....	23
2.6. BufferData.llb	26
2.7. APIManagement.llb – Programming interface controls	26
3. IMPLEMENTATION DETAILS.....	27
3.1. COMMUNICATION LAYERS.....	27
3.2. SAMPLE APPLICATION	28
3.2.1. BLOCK DIAGRAM	29
4. INTERFACE LIBRARY SECTIONS.....	34
4.1. GeneralControlStatus.llb	34
4.1.1. GetCameraHealthStatus.vi	34

4.1.2.	GetCameraType.vi	38
4.1.3.	GetGeneral.vi	42
4.1.4.	GetTemperatures.vi	49
4.1.5.	InitiateSelfTestProcedure.vi	51
4.1.6.	ResetSettingsToDefault.vi	54
4.1.7.	GetCameraName.vi	56
4.1.8.	GetCameraSetup.vi	56
4.1.9.	RebootCamera.vi	58
4.1.10.	SetCameraSetup.vi	59
4.1.11.	SetTimeouts.vi	60
4.1.12.	GetFanControlStatus.vi	61
4.1.13.	SetFanControlStatus.vi	62
4.1.14.	ErrorManager.vi	63
4.1.15.	FormatHWDESC.vi	65
4.1.16.	FormatSoftwareDesc.vi	66
4.2.	Sensor.Ilb	67
4.2.1.	GetADCOperation.vi	67
4.2.2.	GetBinning.vi	68
4.2.3.	GetConversionFactor.vi	70
4.2.4.	GetCoolingSetpointTemperature.vi	72
4.2.5.	GetDescription.vi	74
4.2.6.	GetDescriptionEX.vi	85
4.2.7.	GetDoubleImageMode.vi	88

4.2.8.	GetIRSensitivity.vi	90
4.2.9.	GetOffsetMode.vi	91
4.2.10.	GetPixelRate.vi	93
4.2.11.	GetROI.vi	94
4.2.12.	GetSensorFormat.vi	96
4.2.13.	GetSizes.vi	98
4.2.14.	SetADCOperation.vi	100
4.2.15.	SetBinning.vi	101
4.2.16.	SetConversionFactor.vi	103
4.2.17.	SetCoolingSetpointTemperature.vi	104
4.2.18.	SetDoubleImageMode.vi	105
4.2.19.	SetIRSensitivity.vi	106
4.2.20.	SetOffsetMode.vi	108
4.2.21.	SetPixelRate.vi	110
4.2.22.	SetROI.vi	111
4.2.23.	SetSensorFormat.vi	112
4.2.24.	GetNoiseFilterMode.vi	114
4.2.25.	SetNoiseFilterMode.vi	116
4.2.26.	GetLookupTableInfo.vi	117
4.2.27.	GetActiveLookupTable.vi	119
4.2.28.	SetActiveLookupTable.vi	120
4.2.29.	GetHWIOSignalCount.vi	121
4.2.30.	GetHWIOSignalDescriptor.vi	122

4.3.	TimingControl.llb	124
4.3.1.	ForceTrigger.vi	124
4.3.2.	GetBusyStatus.vi	126
4.3.3.	GetCOCRunTime.vi	127
4.3.4.	GetDelayExposureTime.vi	129
4.3.5.	GetDelayExposureTimeTable.vi	131
4.3.6.	GetExpTrigSignalStatus.vi	133
4.3.7.	GetFPSExposureMode.vi	134
4.3.8.	GetPowerDownMode.vi	135
4.3.9.	GetTrigger.vi	137
4.3.10.	GetUserPowerDownTime.vi	140
4.3.11.	SetDelayExposureTime.vi	141
4.3.12.	SetDelayExposureTimeTable.vi	143
4.3.13.	SetFPSExposureMode.vi	145
4.3.14.	SetPowerDownMode.vi	146
4.3.15.	SetTrigger.vi	148
4.3.16.	SetUserPowerDownTime.vi	150
4.3.17.	GetModulationMode.vi	151
4.3.18.	SetModulationMode.vi	153
4.3.19.	GetFrameRate.vi	154
4.3.20.	SetFrameRate.vi	155
4.3.21.	GetImageTiming.vi	157
4.3.22.	GetHWIOSignal.vi	158

4.3.23.	SetHWIOSignal.vi	161
4.3.24.	GetHWIOSignalTiming.vi.....	162
4.3.25.	SetHWIOSignalTiming.vi	164
4.3.26.	GetCMOSLineTiming.vi	165
4.3.27.	SetCMOSLineTiming.vi.....	167
4.3.28.	GetCMOSLineExposureDelay.vi	168
4.3.29.	SetCMOSLineExposureDelay.vi.....	169
4.3.30.	GetHWLEDSignal.vi	170
4.3.31.	SetHWLEDSignal.vi.....	170
4.4.	Storage.llb	172
4.4.1.	ClearRAMSegment.vi	172
4.4.2.	GetActiveRAMSegment.vi	173
4.4.3.	GetCameraRAMSize.vi	174
4.4.4.	GetCamRAMSegmentSize.vi	175
4.4.5.	SetActiveRAMSegment.vi	176
4.4.6.	SetCamRAMSegmentSize.vi.....	177
4.5.	RecordingControl.llb	179
4.5.1.	ArmCamera.vi	179
4.5.2.	GetAcquireEnableSignalStatus.vi.....	180
4.5.3.	GetAcquireMode.vi.....	181
4.5.4.	GetRecorderSubmode.vi	182
4.5.5.	GetRecordingState.vi	183
4.5.6.	GetStorageMode.vi.....	185

4.5.7.	GetTimeStampMode.vi.....	186
4.5.8.	SetAcquireMode.vi	187
4.5.9.	SetDateTime.vi.....	189
4.5.10.	SetRecorderSubmode.vi	190
4.5.11.	SetRecordingState.vi.....	192
4.5.12.	SetStorageMode.vi	194
4.5.13.	SetTimeStampMode.vi	196
4.6.	BufferData.llb	198
4.6.1.	GetBitAlignment.vi.....	198
4.6.2.	GetImageSegmentSettings.vi.....	200
4.6.3.	GetNumberOfImagesInSegment.vi.....	202
4.6.4.	SetBitAlignment.vi	204
4.6.5.	WriteHotPixelList.vi	205
4.6.6.	ReadHotPixelList.vi	206
4.6.7.	ClearHotPixelList.vi	208
4.6.8.	GetHotPixelCorrectionMode.vi	209
4.6.9.	SetHotPixelCorrectionMode.vi	210
4.6.10.	GetSegmentStructure.vi	211
4.6.11.	GetInterfaceOutputFormat.vi.....	212
4.6.12.	SetInterfaceOutputFormat.vi	213
4.7.	APIManagement.llb	215
4.7.1.	AddBuffer.vi (**Obsolete – Use AddBufferEx.vi for new development)	215
4.7.2.	AddBufferEX.vi.....	216

4.7.3.	AllocateBuffer.vi	218
4.7.4.	CancellImages.vi	220
4.7.5.	CheckDeviceAvailability.vi	221
4.7.6.	CloseCamera.vi	222
4.7.7.	FreeBuffer.vi	223
4.7.8.	GetBufferStatus.vi.....	224
4.7.9.	GetImage.vi (**Obsolete – Use GetImageEx.vi for new development**)	225
4.7.10.	GetImageEX.vi.....	227
4.7.11.	GetImageBuffer.vi.....	229
4.7.12.	GetPendingBuffer.vi	230
4.7.13.	OpenCamera.vi	231
4.7.14.	OpenCameraEx.vi.....	232
4.7.15.	CamLinkSetImageParameters.vi	235
4.7.16.	GetTransferParameters.vi	236
4.7.17.	SetTransferParameters.vi	239
4.7.18.	SetTransferParametersAuto.vi	241
4.7.19.	Cleanup.vi	243
4.7.20.	GetInfoString.vi.....	243
4.7.21.	InitLibrary.vi	244
4.7.22.	Reset_SDK_Lib.vi	245
4.7.23.	WaitForBuffer.vi	246
4.7.24.	Handle_Edge_Pixelrate.vi	248
4.7.25.	EnableSoftROIMode.vi.....	249

5. ERROR / WARNING CODES 251

This document describes the commands for controlling the pco.camera series from within the National Instruments LabVIEW environment. Further explanations appear as needed for the commands, settings and mode configurations.

1. General

- Conventions in this manual
- Hardware elements and interface connectors

1.1. Conventions

The following typographic conventions are used in this manual:

bold: get camera type

Functions, procedures or modes used

[words in brackets]: [run]

Possible values or “states” of the described functions

ALL CAPITAL WORDS: TRUE

Logical or boolean values such as TRUE, FALSE, ON, OFF, 0, 1, RISING, FALLING, HIGH, LOW

<words in arrows>: <acq enbl>

Names of hardware input / output signals

1.2. Hardware elements, interface connectors

All of the pco.camera designs have some input and output capability for triggering and synchronization with other equipment. In some models, these inputs and outputs are configured using the DIP switches on the back of the pco.camera power supply:

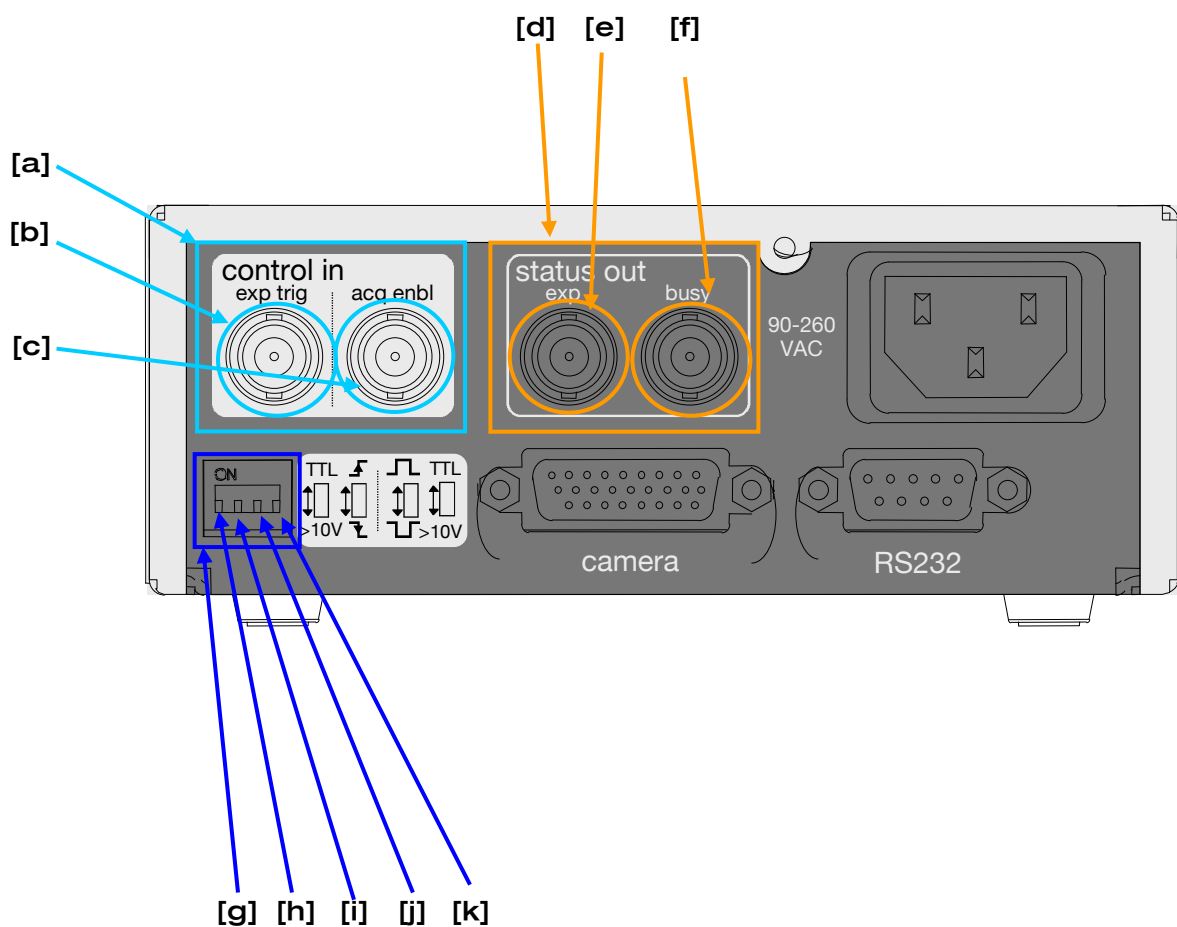






Figure 1: View of back panel of pco.power

legend:

- [a] <control in> - general input for external control signals, BNC plugs
- [b] <exp trig> - external exposure trigger input
- [c] <acq enbl> - external acquire enable input
- [d] <status out> - general status output signals, BNC plugs
- [e] <exp> - exposure output signal
- [f] <busy> - busy output signal
- [g] DIP switch, which sets polarity, HIGH and LOW levels and level of voltages
- [h] [TTL] or [>10V] selects voltage level either TTL = 5V or larger than 10V
- [i]  or  trigger edge selection for <exp trig> input, either rising or falling edge
- [j]  or  trigger level selection for <acq enbl> input, HIGH or LOW active
- [k] [TTL] or [>10V] selects voltage level either TTL = 5V or larger than 10V

In other camera models, such as the pco.edge and the pco.dimax series, the I/O is configured in software using the HWIO functions described below.

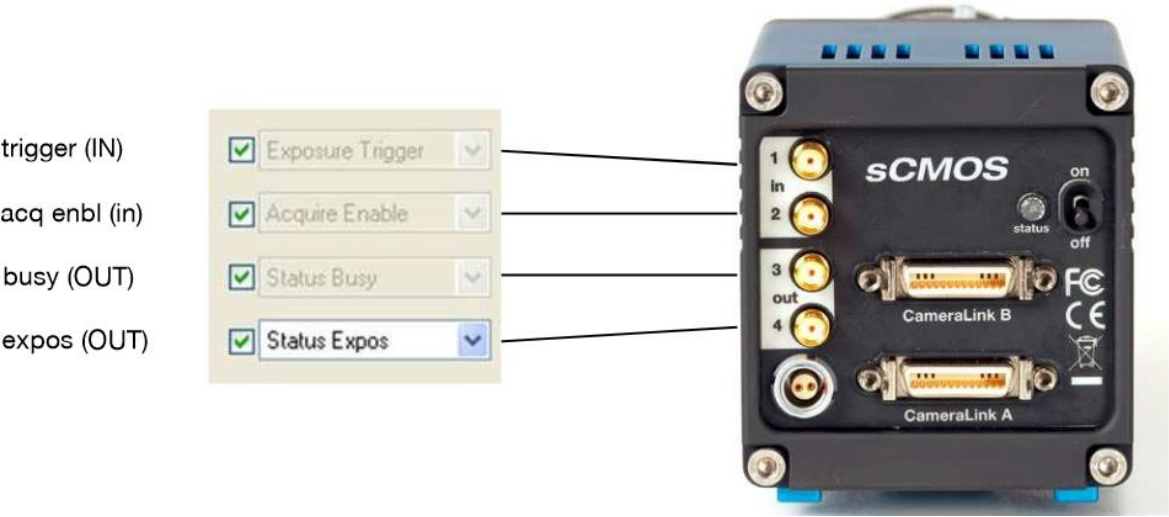
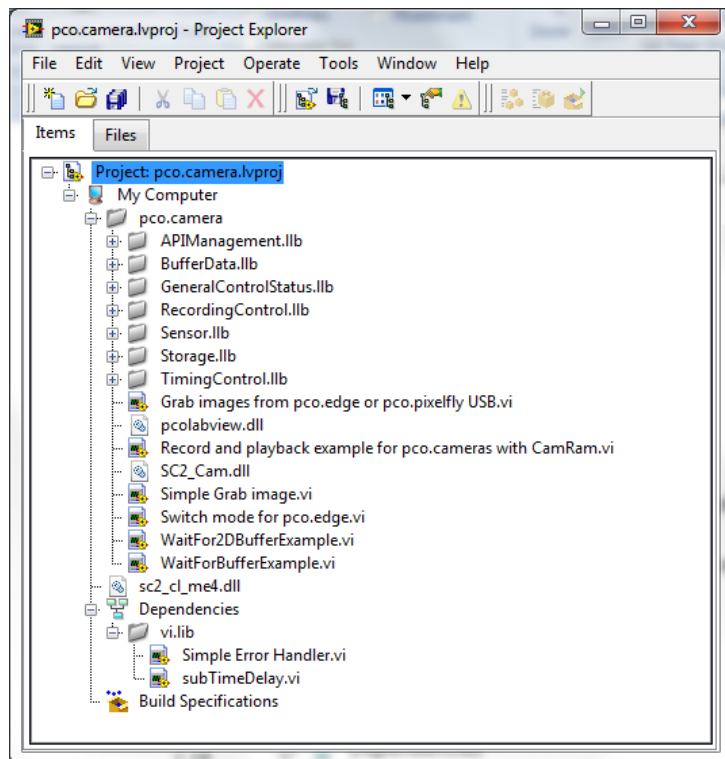


Figure 2: pco.edge signal connections

2. Overview and function description

The complete LabVIEW interface is comprised of LabVIEW VI libraries and the Dynamic Link Libraries that connect LabVIEW to the camera driver. All of the relevant files contained in a single LabVIEW project.



The LabVIEW interface consists of the following files:

- VI Libraries: APIManagement.llb, BufferData.llb, GeneralControlStatus.llb, RecordingControl.llb, Sensor.llb, Storage.llb, TimingControl.llb, Example VI's
- Dynamic Link Libraries: pcolabview.dll, SC2_Cam.lib, SC2_1394.dll, SC2_cl_me4.dll, etc.

There are libraries of virtual instruments (VI's) used for camera control and image acquisition, and lower-level dynamic link library files. Most of the VI's call functions in the dynamic link library SC2_Cam.dll, which should reside in your application directory. The SC2_Cam.dll in turn accesses lower interface libraries, which should be installed during the driver installation. These interface libraries are typically named sc2_XXXX.dll, e.g. sc2_cl_ME4.dll or sc2_1394.dll. Another DLL, pcolabview.dll, handles memory management between the LabVIEW environment and the DLL's. Other dll's may be added with new releases to accommodate different interface types.

As new camera functionality and camera models are added to the pco.camera family, the interface dll's may be updated to allow access to them. It may be necessary to update the DLL's in the LabVIEW project. This is done by obtaining the latest pco.runtime from www.pco-tech.com or www.pco.de, and copying the latest DLL's from the runtime package (e.g. SW_PCORUNTIMEWIN_117.zip) to the LabVIEW project folder.

Driver versions: 32 vs 64 bit.

Beginning with version 2009, LabVIEW is available as both a 32 or 64 bit application. Compatible drivers are available for both versions. It is necessary to use a consistent software stack to get the LabVIEW interface to work correctly. This means that if you are using a 64-bit device driver for the camera interface, then you must use the 64 bit LabVIEW driver and 64-bit LabVIEW. Your LabVIEW license will allow you to use either version of LabVIEW.

The VI's are grouped in libraries by function.

- Camera (General)
- Image Sensor
- Timing
- Storage
- Recording
- Buffer control
- API-Management

2.1. GeneralControlStatus.llb – Camera control

This library contains general functions to control the camera and to request information about the camera. For example:

- Request camera type, hardware/firmware version, serial number, interface type
- Request camera status (warnings, errors etc.)
- Reset all settings to default values
- Initiate self test procedure
- Get camera / power supply temperature

2.2. Sensor.llb - Image sensor specifics

This group contains complete image sensor control instructions and instructions to request information about the sensor. Some of these are:

- Get Camera description: sensor type, standard resolution, extended resolution, dynamic resolution (bit), delay and exposure times...
- Set/request sensor format: [standard] / [extended].
- Set/request ROI settings.
- Set/request binning settings.
- Set/request pixel rate (frequency for shifting the pixels out of the sensor shift registers).
- Set/request conversion factor (gain) settings.
- Set/request double image mode (expose two images one after another immediately).
- Set/request ADC mode (use one or two ADCs for digitizing the pixel data of the sensor).
- Set/request IR sensitivity setting (ON/OFF).
- Set/request cooling set point temperature.
- Set/request Offset Mode.

2.3. TimingControl.llb – Image timing

This group contains all available commands for control of imaging process timing:

- Set / request delay and exposure time (timebase, timetable) for taking images.
- Set / request trigger mode for exposures: [auto trigger], [force trigger], [extern edge triggered], [extern exposure pulse trigger]⁽¹⁾. Controls the usage of the <exp trig> control input. See below for a detailed description of the trigger modes.
- Force trigger: this software command starts an exposure if the trigger mode is in the state [auto trigger], [force trigger] or [extern edge triggered]. If in [extern exposure pulse trigger] mode nothing happens.

- Request busy status: A trigger is ignored if the camera is still busy (exposure or readout). In case of [force trigger] command, the user may request the camera's busy status in order to generate a valid [force trigger] command.
- Set / request power down time (threshold value, which becomes available in case of exposure times longer than 1s)
- Read control input (<exp trig>): read TRUE or FALSE level of external control input⁽²⁾ (<control in>).

Notes:

- (1) Edge type (FALLING edge / RISING edge) as well as the electrical sensitivity (trigger level) are selected by DIP switches at the power supply unit near the trigger input(<control in>). In double image mode, the first exposure time is affected by the trigger commands. The duration of the second exposure is always given by the readout time of the first image.
- (2) If the DIP switch shows a RISING edge, then the HIGH level signal is TRUE and the LOW level signal is FALSE. If the DIP switch shows a FALLING edge, then the HIGH level signal is FALSE and the LOW level signal is TRUE.

The following table shows how the different trigger modes work:

Trigger mode	Operation Description
auto trigger	A new image exposure is automatically started as soon as possible compared to the readout of an image. If a CCD is used and images are taken in sequence, then exposures and sensor readout are started simultaneously.
software trigger	An exposure can only be started by a force trigger command.
extern exposure & software trigger	A delay / exposure sequence is started at the RISING or FALLING edge ⁽¹⁾ of the trigger input (<control in>) or by a [force trigger] command.
extern exposure control	The exposure time is defined by pulse length at the trigger input (<control in>). The delay and exposure time values defined by the set / request delay and exposure command are ineffective.

2.4. Storage.llb – Camera memory management

For camera models with on-board memory (camera RAM or camRAM) this library contains all commands needed for controlling the memory and storage process.

The total camera memory is divided into four segments (similar to partitions on hard discs). The VI's in this library can be used to configure the camRAM segments. For example, use these VI's to:

- Request RAM size (pages) and page size (pixels)
- Request / set RAM segment size in pages
- Clear RAM segment
- Get / set active RAM segment

Note:

A consistency check (in order to avoid buffers that overlap) must be performed by the application software!



Each segment also contains information about the image settings (ROI / binning etc.) for the images stored within this segment (all images must have the same format).

2.5. RecordingControl.llb – Image recording controls

- Set / request storage mode: [recorder mode] / [FIFO buffer mode] (see insert box 2.5.1 for further explanations)
- Set / request recorder submode: [sequence] / [ring buffer] (see insert box 2.5.2 for further explanations)
- Set / request recording state: [run] / [stop] (see insert box 2.5.3 for further explanations)
- Arm: prepare camera for recording command
This function is necessary before a new recording (**set recording** = [run]) command is released. This function takes the delay, exposure, triggering, recorder mode (etc.) settings, compiles them and prepares the camera to start immediately when a start of recording (**set recording** = [run]) is performed.
- Set / request acquire mode: [auto] / [external], controls the usage of the <acq enbl> control input
 - [auto]: the external control input <acq enbl> is ignored
 - [external]: the external control input <acq enbl> is a static enable signal of images. If this input is TRUE, then exposure triggers are accepted and images are taken. If this signal is set FALSE, then all exposure triggers are ignored and the sensor readout is stopped.
- Read control input (<acq enbl>): read TRUE or FALSE level of external control input⁽¹⁾ (<control in>)
- Set date / time
- Set / request timestamp mode

Notes:

Active (TRUE) level (LOW/HIGH) as well as the electrical sensitivity is selected by DIP switches at the power supply unit near the acquire enable input(<acq enbl>).

- (1) If the DIP switch shows  then the HIGH level signal is TRUE and the LOW level signal is FALSE. If the DIP switch shows  then the HIGH level signal is FALSE and the LOW level signal is TRUE.

Box 2.5.1

recorder mode	FIFO buffer mode
<ul style="list-style-type: none"> • images are recorded and stored within the internal camera memory (camRAM) 	<ul style="list-style-type: none"> • all images taken are transferred to the PC in chronological order

<ul style="list-style-type: none"> • “live view” transfers the most recent image to the PC (for viewing / monitoring) • indexed or total image readout after the recording has been stopped 	<ul style="list-style-type: none"> • camera memory (camRAM) is used as a huge FIFO buffer to bypass short bottlenecks in data transmission. If buffer overflows, the oldest images are overwritten. In FIFO buffer mode, images are sent directly to the PC interface (FireWire, USB ...) like a continuous data stream. Synchronization is done with the interface.
---	--

Box 2.5.2

recorder submode: sequence	recorder submode: ring buffer
<ul style="list-style-type: none"> • Recording is stopped when the allocated buffer is full. 	<ul style="list-style-type: none"> • Camera records continuously into ring buffer. If the allocated buffer is full, the older images are overwritten. Recording is stopped by software command.

Box 2.5.3

Recording: [run] / [stop]
<p>The recording command controls the camera status. If the recording state is [run], images can be released by exposure trigger and acquire enable. If the recording state is [stop] all image readout or exposure sequences are stopped and the sensors (CCDs or CMOS) are running in a special idle mode to prevent dark charge accumulation.</p> <p>The recording state has the highest priority compared to functions like acquire enable or exposure trigger.</p> <p>The recording state is started by:</p> <ul style="list-style-type: none"> • software command: Set recording = [run] <p>The recording state is stopped by:</p>

- powering on the camera
- software command: **Set recording** = [stop]
- software command: Reset all settings to default values.
- in recorder submode = [sequence], if the buffer overflows.

2.6. BufferData.llb

- Request image settings for this segment (ROI, binning, horizontal x vertical resolution)
- Request number of images in segment

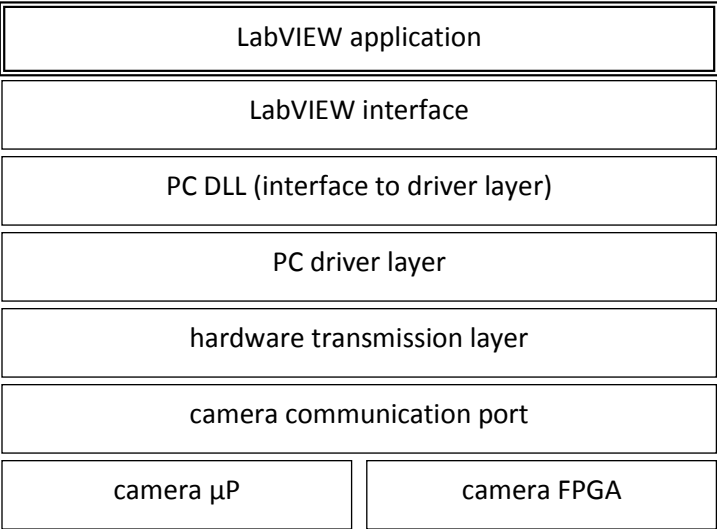
The image readout is part of the API-management commands. If the camera is in recording state the PCO_AddBuffer command must be used. If the camera is not in recording state, the PCO_GetImage command must be used.

2.7. APIManagement.llb – Programming interface controls

- Open and close the camera device
- Buffer management (allocate, free, add buffer, get status) and image access
- Device availability during runtime

3. Implementation Details

3.1. Communication Layers



The application software running on the PC is able to send commands to the camera as well as request status information from the camera. There is also a channel for transmitting image data.

The interface links the LabView application software to the camera device driver layer. Commands sent to the driver should be common for all camera versions as well as for all types of interfaces (FireWire, USB etc.). Thus, the driver converts the commands to the used hardware port.

pco.labview

A common software interface is provided for all communication interface types, so typically no code change is needed to communicate with different camera models or different interfaces.

The communication port, that is the path from the PC driver layer down, separates the data path into channels for commands, status messages and image data.

3.2. Sample application

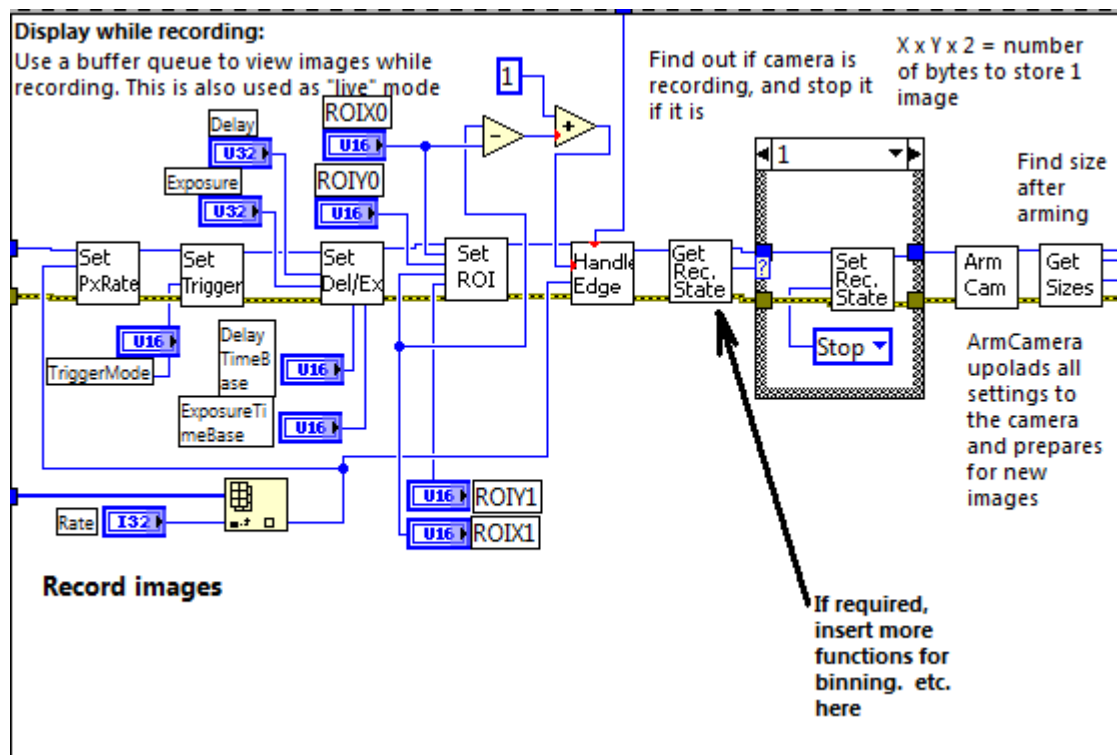
A basic sample application is provided with the driver libraries, to illustrate the camera modes of operation. Developers can use this as a basis for further development, by inserting library functions into the sample application at the appropriate points.

The sample program illustrates how to obtain images from the pco.camera while the camera is recording, through the use of the buffer queue and the buffer events.

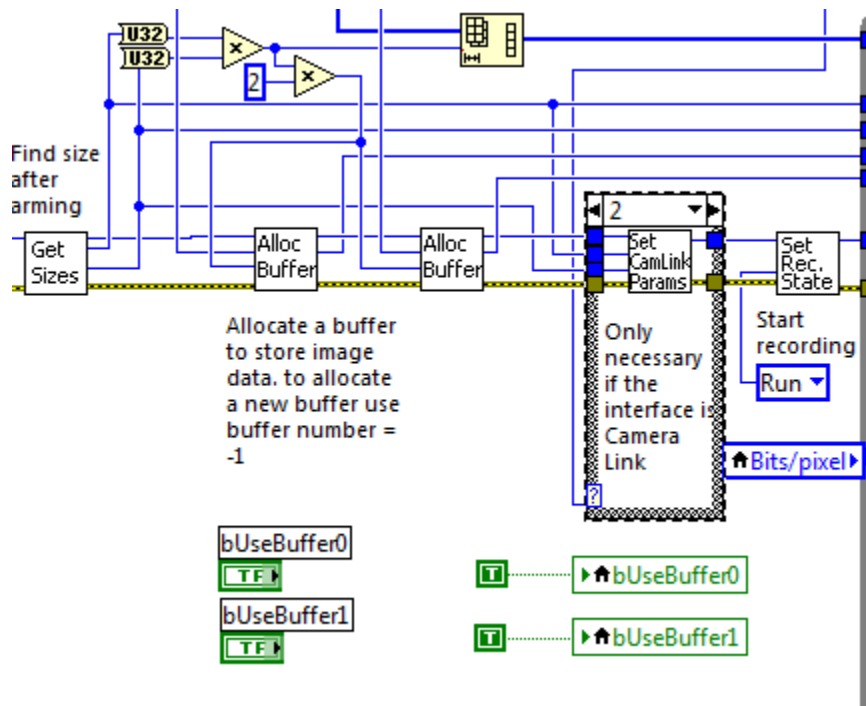
Grab images from pco.edge or pixelfly USB.vi

This VI is a short program to demonstrate the fundamentals of control and image acquisition with the pco.camera series. The program records a sequence of images, with "live" updating, until the "StopRecording" button is pressed. Though the title specifically references the pco.edge and pco.pixelfly, any pco.camera can be controlled from this VI. Other camera models with camRAM require additional code to retrieve recorded images from camRAM. Other code examples are provided to illustrate this.

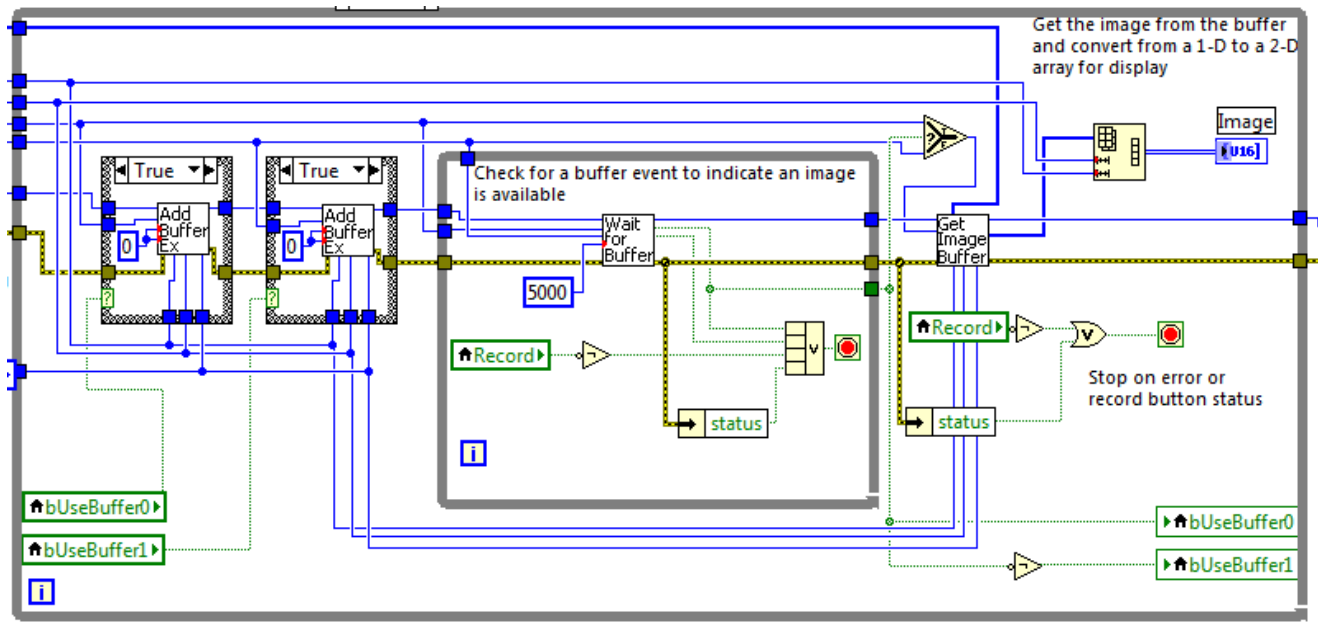




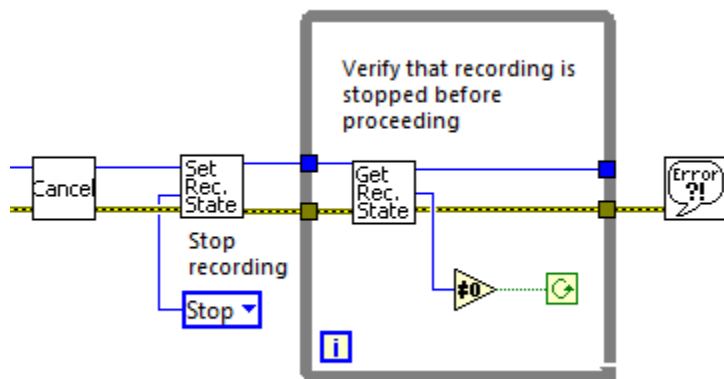
Once the camera is armed, the image size can be queried using GetSizes.vi. The size of the array is then modified to fit the image. A buffer is allocated for viewing images while recording. Setting the recording state to RUN with the SetRecordingState.vi starts the recording process.



While recording, images can be obtained from the camera by adding a buffer to a queue to receive them, using the AddBufferEx.vi. Two buffers can be maintained in a queue by adding one buffer as the other one gets filled. WaitForBuffer.vi determines if there is an image available, and GetImageBuffer.vi retrieves it from the buffer and places it in a 1-D array. The array is reshaped to 2-D for display

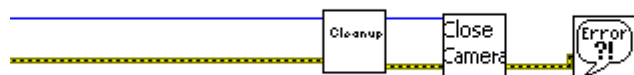


The recording process is halted by setting the recording state to “Stop”. Any buffers used are removed from the queue with CancellImages.vi.



After all camera operations are complete, buffer memory is de-allocated by Cleanup.vi, and the camera is closed using the CloseCamera.vi.

Quit: Free up any allocated buffers and close the camera communication channel



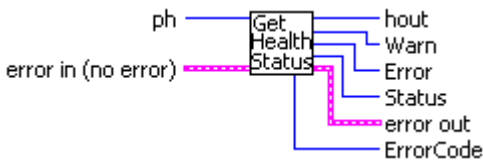
4. Interface library sections

4.1. GeneralControlStatus.llb

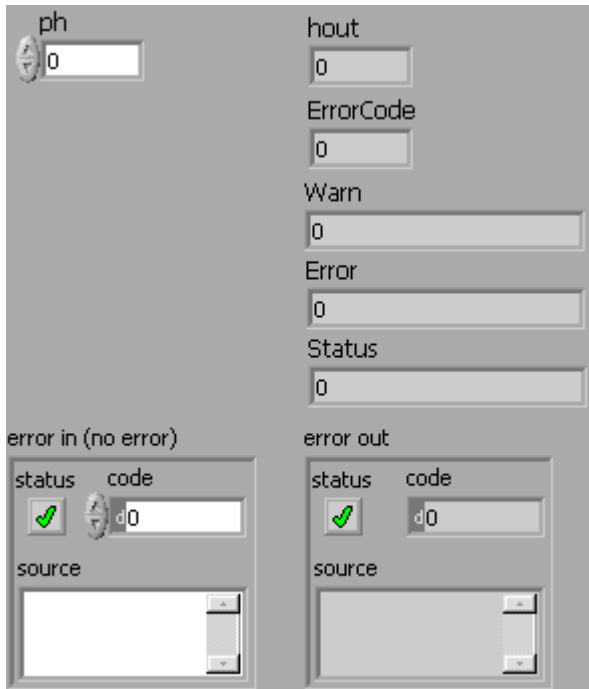
4.1.1. GetCameraHealthStatus.vi

Returns information on the operational status of the camera, including any error conditions that may exist.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



Warn Indicates a potential problem, but one which is not serious enough to be considered an error. The warnings can be interpreted bit-wise as follows:

0x00000001 Power Supply Voltage Range

0x00000002 Power Supply Temperature

0x00000004 Camera temperature (board temperature / FPGA temperature)

0x00000008 Image Sensor temperature (for cooled camera versions only)

Multiple error conditions can exist at the same time. For example, if there is a warning about the power supply temperature and the image sensor temperature, the code would be 0x0000000A.



Error Indicates an error condition in the camera. The code can be interpreted bit-wise as follows:

0x00000001 Power Supply Voltage Range

0x00000002 Power Supply Temperature

0x00000004 Camera temperature (board temperature / FPGA temperature)

0x00000008 Image Sensor temperature (for cooled camera versions only)

0x00010000 Camera Interface failure

0x00020000 Camera RAM module failure

0x00040000 Camera Main Board failure

0x00080000 Camera Head Boards failure

Multiple error conditions can exist at the same time. For example, if the camera interface and the main board both have errors, the code would be 0x00050000

U32

Status Indicates the general status of the camera. The code can be interpreted bit-wise as follows:

0x00000001 Default State:

- Bit set: Settings were changed since power up or reset.
- Bit cleared: No settings changed, camera is in default state.

0x00000002 Settings Valid:

- Bit set: Settings are valid (i.e. last "Arm Camera" was successful and no settings were changed since 'Arm camera', except exposure time).
- Bit cleared: Settings were changed but not yet checked and accepted by 'Arm Camera' command.

0x00000004 Recording State:

- Bit set: Recording state is on.

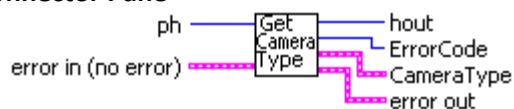
- Bit cleared: Recording state is off.

Multiple status indicators may be present. For example, if the settings have been changed, and the last setting was valid, the code would be 0x00000003

4.1.2. GetCameraType.vi

Returns information about the type of camera referenced by the handle input. This information includes the camera type, subtype, serial number, along with version information for the hardware and firmware.

Connector Pane



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode Error code returned by the driver.



CameraType Cluster of information about the camera versions.



Size Size in bytes of all the information returned by the driver.

U16 **CamType** Type of camera referenced by the handle. Valid types:

0x100 - pco.1200hs

0x200 - pco.1300

0x220 - pco.1600

0x240 - pco.2000

0x260 - pco.4000

Higher numbers are reserved for future use

U16 **CamSubType** Sub-type of the camera referenced by the handle

U32 **SerialNumber** Serial number of the camera head

U32 **HWVersion** Hardware version, coded as two hexadecimal words. The MS word is the major revision number. LS word is the minor revision number

e.g.:

0x00020001 - Version 2.01


More details are available in the HardwareVersion cluster

U32 **FWVersion** Firmware version, coded as two hexadecimal words. The MS word is the major revision number. LS word is the minor revision number


e.g.:

0x00020001 - Version 2.01

More details are available in the FirmwareVersion cluster

 **InterfaceType** Physical layer interface for this camera connection.


 **HardwareVersion** Detailed description of hardware found in the camera system


 **BoardNum** Number of boards found in the system


 **HWVersions** Detailed hardware information for each board



 **BoardName** Text description of board

 **BatchNumber** Code describing batch that this board is from.

 **MinorRevision** Minor hardware revision code for this board, e.g. if version is 2.01, minor revision is 1


 **MajorRevision** Major hardware revision code for this board, e.g. if version is 2.01, major revision is 2

 **Variant** If there is a special variant for this board, a code

for this variant will appear here



FirmwareVersion



DeviceNum Number of devices (processors or gate arrays) found in the system




FWVersions Detailed firmware information for each device







DeviceName Text description of device



MinorRevision Minor firmware revision code for this device, e.g. if version is 2.01, minor revision is 1



MajorRevision Major firmware revision code for this device, e.g. if version is 2.01, major revision is 2

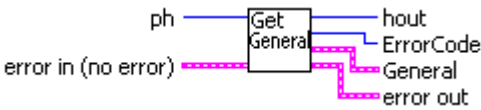


Variant If there is a special variant for this device, a code for this variant will appear here

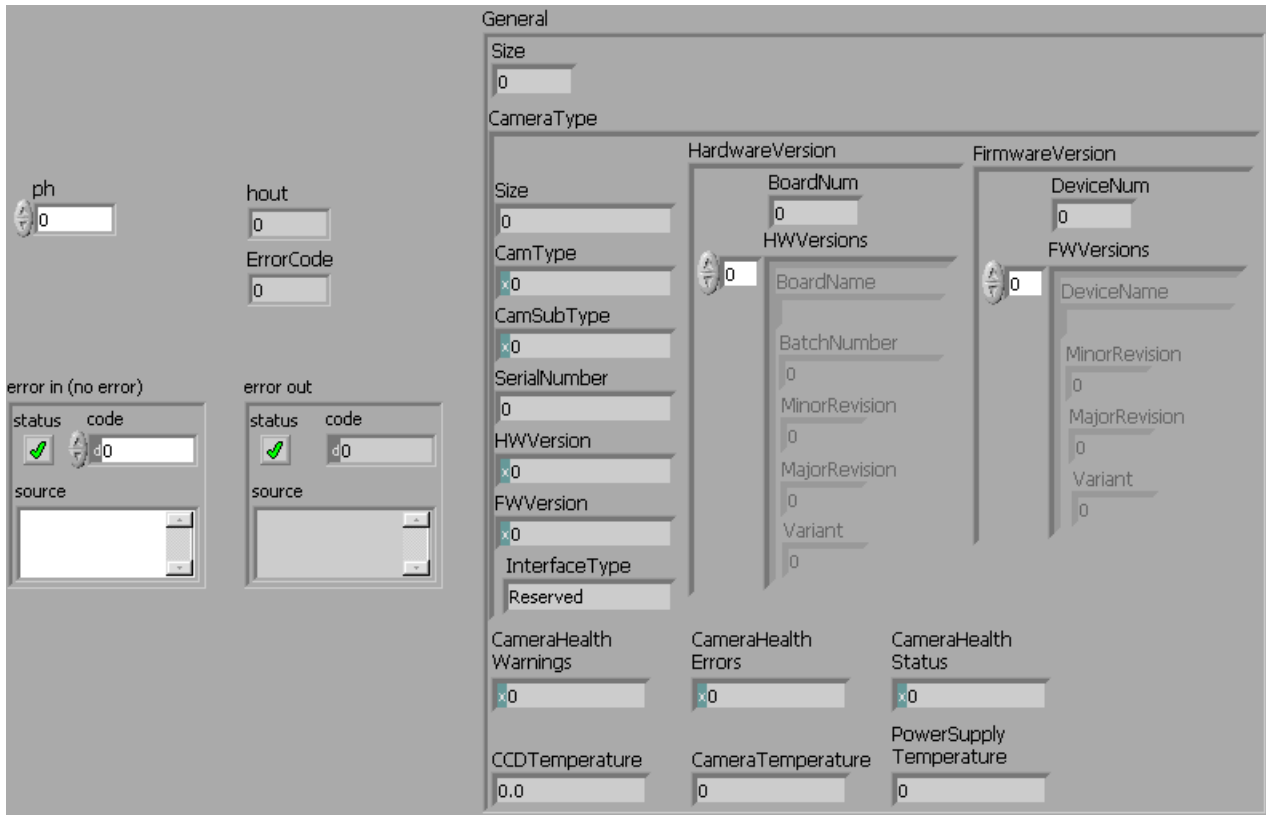
4.1.3. GetGeneral.vi

Returns information on the type, error status and physical state of the camera.

Connector Pane



Front Panel



Controls and Indicators

 **ph** Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode Error number returned by driver



General Camera information cluster



Size Size in bytes of all the information returned by the driver.



CameraType



Size Size in bytes of all the information returned by the driver.



CamType Type of camera referenced by the handle. Valid types:

0x100 - pco.1200hs

0x200 - pco.1300

0x220 - pco.1600

0x240 - pco.2000

0x260 - pco.4000

Higher numbers are reserved for future use

U16 **CamSubType** Sub-type of the camera referenced by the handle

U32 **SerialNumber** Serial number of the camera head

U32 **HWVersion** Hardware version, coded as two hexadecimal words. The MS word is the major revision number. LS word is the minor revision number

e.g.:

0x00020001 - Version 2.01

More details are available in the HardwareVersion cluster


U32 **FWVersion** Firmware version, coded as two hexadecimal words. The MS word is the major revision number. LS word is the minor revision number


e.g.:

0x00020001 - Version 2.01

More details are available in the FirmwareVersion cluster

U16 **InterfaceType** Physical layer interface for this camera connection.


**HardwareVersion** Detailed description of hardware found in the camera system


**BoardNum** Number of boards found in the system


**HWVersions** Detailed hardware information for each board




**BoardName** Text description of board

**BatchNumber** Code describing batch that this board is from.

**MinorRevision** Minor hardware revision code for this board, e.g. if version is 2.01, minor revision is 1

**MajorRevision** Major hardware revision code for this board, e.g. if version is 2.01, major revision is 2

**Variant** If there is a special variant for this board, a code for this variant will appear here

**FirmwareVersion**

**DeviceNum** Number of devices (processors or gate arrays)

found in the system



FWVersions Detailed firmware information for each device



DeviceName Text description of device



MinorRevision Minor firmware revision code for this device, e.g. if version is 2.01, minor revision is 1



MajorRevision Major firmware revision code for this device, e.g. if version is 2.01, major revision is 2



Variant If there is a special variant for this device, a code for this variant will appear here



CameraHealthWarnings Indicates a potential problem, but one which is not serious enough to be considered an error. The warnings can be interpreted bit-wise as follows:

0x00000001 Power Supply Voltage Range

0x00000002 Power Supply Temperature

0x00000004 Camera temperature (board temperature / FPGA temperature)

0x00000008 Image Sensor temperature (for cooled camera versions only)

Multiple error conditions can exist at the same time. For example, if there is a warning about the power supply temperature and the image sensor temperature, the code would be 0x0000000A.



CameraHealthErrors Indicates an error condition in the camera. The code can be interpreted bit-wise as follows:

0x00000001 Power Supply Voltage Range

0x00000002 Power Supply Temperature

0x00000004 Camera temperature (board temperature / FPGA temperature)

0x00000008 Image Sensor temperature (for cooled camera versions only)

0x00010000 Camera Interface failure

0x00020000 Camera RAM module failure

0x00040000 Camera Main Board failure

0x00080000 Camera Head Boards failure

Multiple error conditions can exist at the same time. For example, if the camera interface and the main board both have errors, the code would be 0x00050000



CameraHealthStatus Indicates the general status of the camera. The code can be interpreted bit-wise as follows:

0x00000001 Default State:

- Bit set: Settings were changed since power up or reset.
- Bit cleared: No settings changed, camera is in default state.

0x00000002 Settings Valid:

- Bit set: Settings are valid (i.e. last “Arm Camera” was successful and no settings were changed since ‘Arm camera’, except exposure time).
- Bit cleared: Settings were changed but not yet checked and accepted by ‘Arm Camera’ command.

0x00000004 Recording State:

- Bit set: Recording state is on.
- Bit cleared: Recording state is off.

Multiple status indicators may be present. For example, if the settings have been changed, and the last setting was valid, the code would be 0x00000003



CCDTemperature Temperature in Celsius of the image sensor



CameraTemperature Temperature in Celsius of the camera head electronics

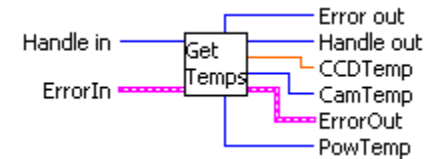


PowerSupplyTemperature Temperature in Celsius of the power supply electronics.

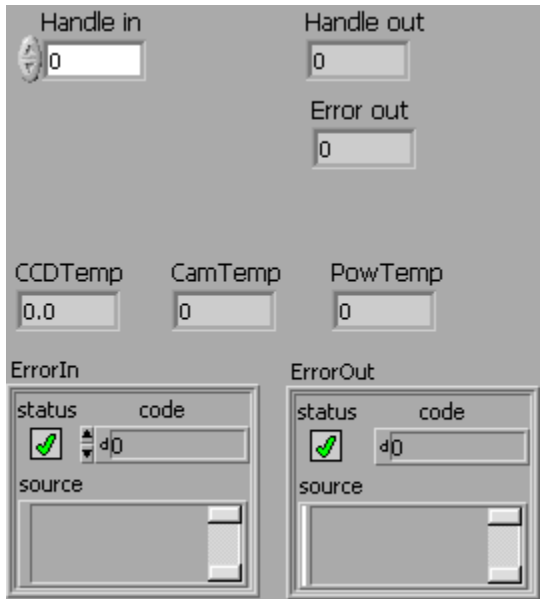
4.1.4. GetTemperatures.vi

Get current sensor, electronics and power supply temperatures.

Connector Pane



Front Panel



Controls and Indicators

U32 Handle in Handle to camera

E31 **ErrorIn** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

I32 Error out



Handle out



CCDTemp Sensor temperature in Celsius



CamTemp Electronics temperature in Celsius



PowTemp Power supply temperature in Celsius



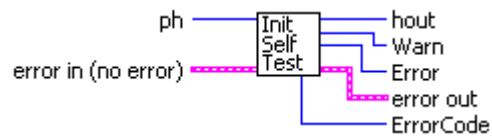
ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

.

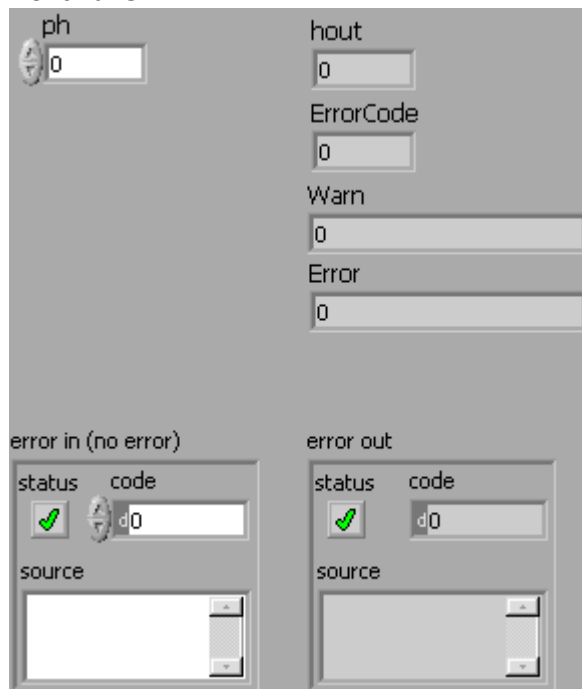
4.1.5. InitiateSelfTestProcedure.vi

Initiates a camera self-test and returns any errors or warnings encountered.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed

in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



Warn Indicates a potential problem, but one which is not serious enough to be considered an error. The warnings can be interpreted bit-wise as follows:

0x00000001 Power Supply Voltage Range

0x00000002 Power Supply Temperature

0x00000004 Camera temperature (board temperature / FPGA temperature)

0x00000008 Image Sensor temperature (for cooled camera versions only)

Multiple error conditions can exist at the same time. For example, if there is a warning about the power supply temperature and the image sensor temperature, the code would be 0x0000000A.



Error Indicates an error condition in the camera. The code can be interpreted bit-wise as follows:

0x00000001 Power Supply Voltage Range

0x00000002 Power Supply Temperature

0x00000004 Camera temperature (board temperature / FPGA temperature)

0x00000008 Image Sensor temperature (for cooled camera versions only)

0x00010000 Camera Interface failure

0x00020000 Camera RAM module failure

0x00040000 Camera Main Board failure

0x00080000 Camera Head Boards failure

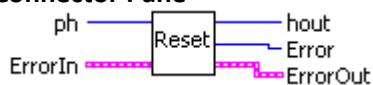
Multiple error conditions can exist at the same time. For example, if the camera interface and the main board both have errors, the code would be 0x00050000

4.1.6. ResetSettingsToDefault.vi

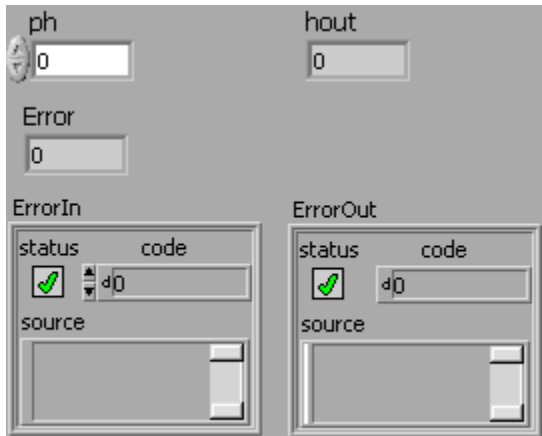
Resets all camera settings to default values. These values are:

Sensor Format:	standard
ROI:	full resolution
Binning:	no binning (1 X 1)
Pixel Rate:	Lowest rate (sensor dependent)
Gain:	Normal gain (if setting available due to sensor)
Double Image Mode:	Off
IR sensitivity:	Off (if setting available due to sensor)
Cooler Setpoint:	Depends on camera model
ADC mode:	Using one ADC
Exposure Time:	10 ms
Delay Time:	0 μ s
Trigger Mode:	Auto Trigger
Recording state:	stopped
Memory Segmentation:	Total memory allocated to first segment
Storage Mode Recorder:	Ring Buffer + Live View on
Acquire Mode:	Auto

Connector Pane



Front Panel



Controls and Indicators



ph



ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout



Error



ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

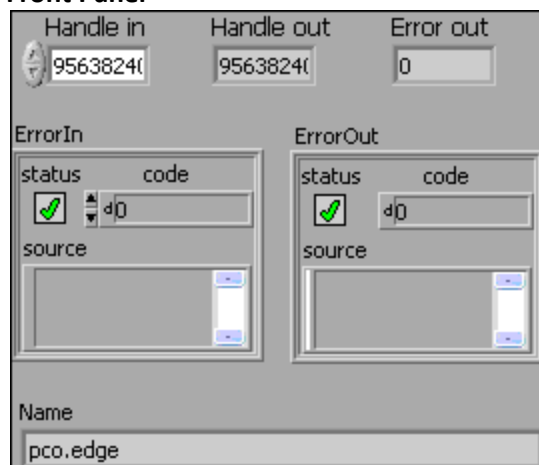
4.1.7. GetCameraName.vi

Retrieve camera name text stored in camera head

Connector Pane



Front Panel



Controls and Indicators



Handle in



ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Error out



Handle out



ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

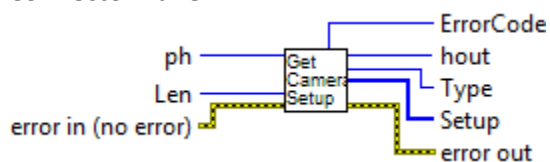


Name Name text stored in camera head

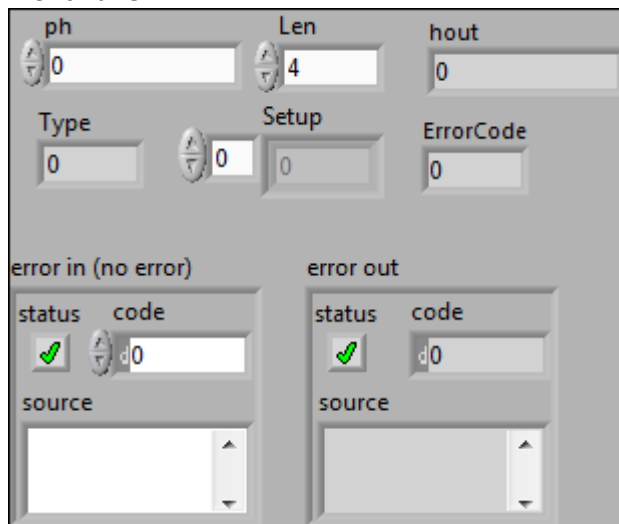
4.1.8. GetCameraSetup.vi

Obtains information on the camera setup, as an array of flags. From these flags, it can be determined, for example, if a pco.edge camera is set in rolling or global shutter mode.








Connector Pane



Front Panel



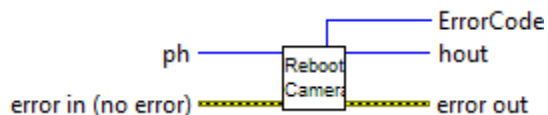
Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called.
-  **Len** Length of the array that holds the camera flags.
-  **hout** Handle output
-  **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
-  **ErrorCode**
-  **Type** Type of setup detected.

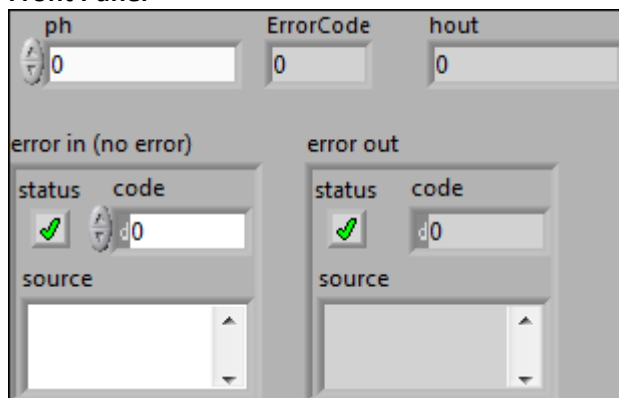
4.1.9.RebootCamera.vi

Re-initializes the camera to its power-up state.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

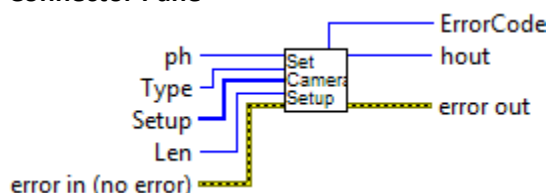


ErrorCode

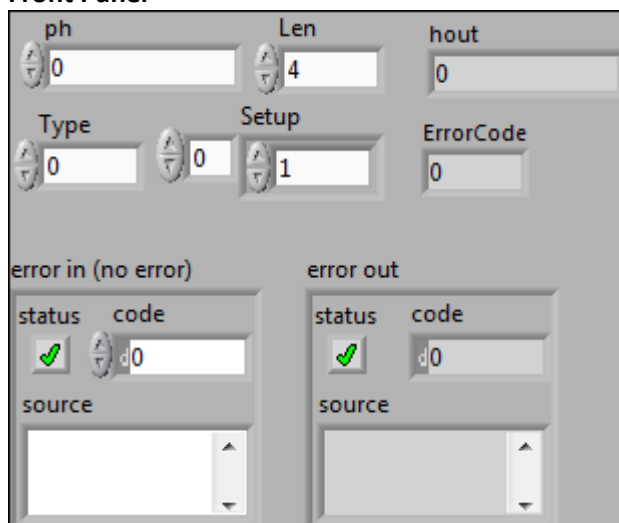
4.1.10. SetCameraSetup.vi

Makes changes to the camera setup, using an array of flags. From these flags, for example, a pco.edge camera can be set in rolling or global shutter mode.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called.



Len Length of the array that holds the camera flags. Default is 4.



Type Type of setup selected.



Setup Array of flags, controlling camera setup.

Setup[0] = 1 : pco.edge rolling shutter mode selected

Setup[0] = 2 : pco.edge global shutter mode selected

Setup[0] = 4 : pco.edge global reset mode selected

U64 **hout** Handle output

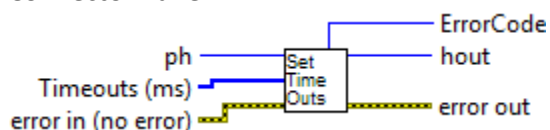
E64 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

I32 **ErrorCode**

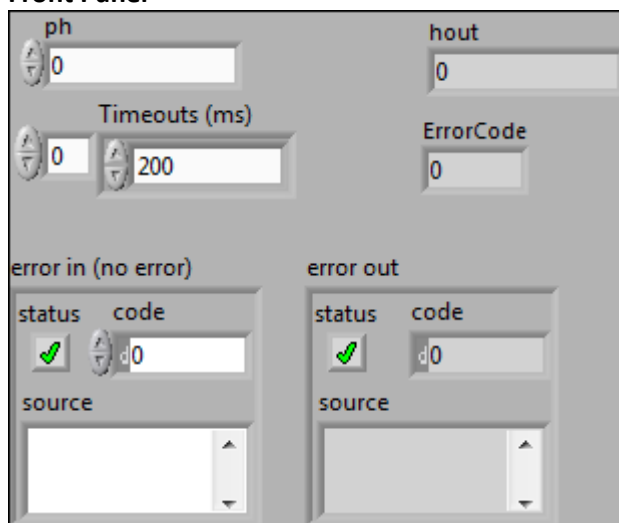
4.1.11. SetTimeouts.vi

Set the time out values for the driver. Separate values are used for command transfer, image transfer, and transfer channel expiry.

Connector Pane



Front Panel



Controls and Indicators

U64 **ph** Handle for the camera

E64 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called.

U32 **Timeouts (ms)** Array of timeout values, in milliseconds.

Timeout[0] = Time to wait while a command is sent

Timeout[1] = Time to wait for an image to be transferred

Timeout[3] =Time to wait for transfer channel to expire



hout Handle output



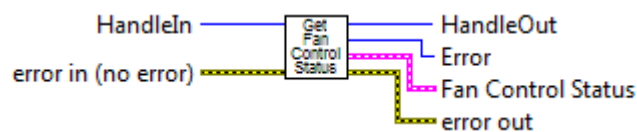
error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode

4.1.12. GetFanControlStatus.vi

Returns information on fan set point and control mode. Indicates if fan is under automatic or manual control.



Front panel of the Get Fan Control Status VI. The panel displays the following controls:

- HandleIn: 0
- Error: 0
- HandleOut: 0
- Fan Control Status: Manual control (selected)
- FanMin: 0
- FanMax: 100
- FanStep: 1
- FanSetPoint: 10
- FanActual: 10
- Reserved: 0
- error in (no error): status (green checkmark), code (0), source (empty)
- error out: status (green checkmark), code (0), source (empty)



HandleIn



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



HandleOut



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



Fan Control Status Contains information about the fan control status in the pco.edge.



Manual control Indicates if the fan is available for manual control

FALSE - Fan is under automatic control
TRUE - Fan can be controlled manually.



FanMin Minimum value for fan speed. This value is a percentage of the maximum fan voltage.



FanMax Maximum value for fan speed. This value is a percentage of the maximum fan voltage.



FanStep Step size for the fan speed set point. This is a percentage of the maximum fan voltage.



FanSetPoint Sets the operating point for the fan. This is a percentage of the maximum fan voltage.



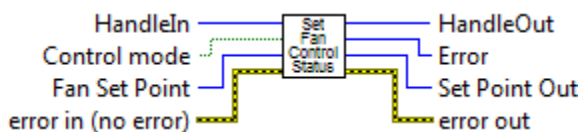
FanActual Actual operating point of the fan, as a percentage of the maximum fan voltage.

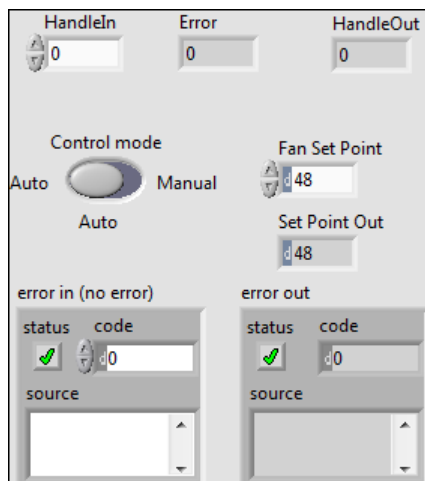


Reserved Reserved for future control options


4.1.13. SetFanControlStatus.vi


Controls the mode of fan operation, and sets the operating point of the fan in manual mode. Only pco.edge cameras with internal fan are supported.






HandleIn


 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.


 **Control mode** Switches the fan control mode from automatic to manual.

FALSE = Automatic (default)
TRUE = Manual

 **Fan Set Point** In manual mode, this sets the fan speed. This input is ignored if the fan is in auto mode. The number represents a percentage of the maximum fan voltage. To find the range of acceptable values, use `GetFanControlStatus.vi`.

HandleOut

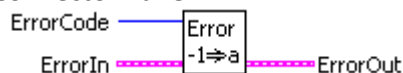
 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

 **Set Point Out** Current fan setpoint. The number represents a percentage of the maximum fan voltage. To find the range of acceptable values, use `GetFanControlStatus.vi`.

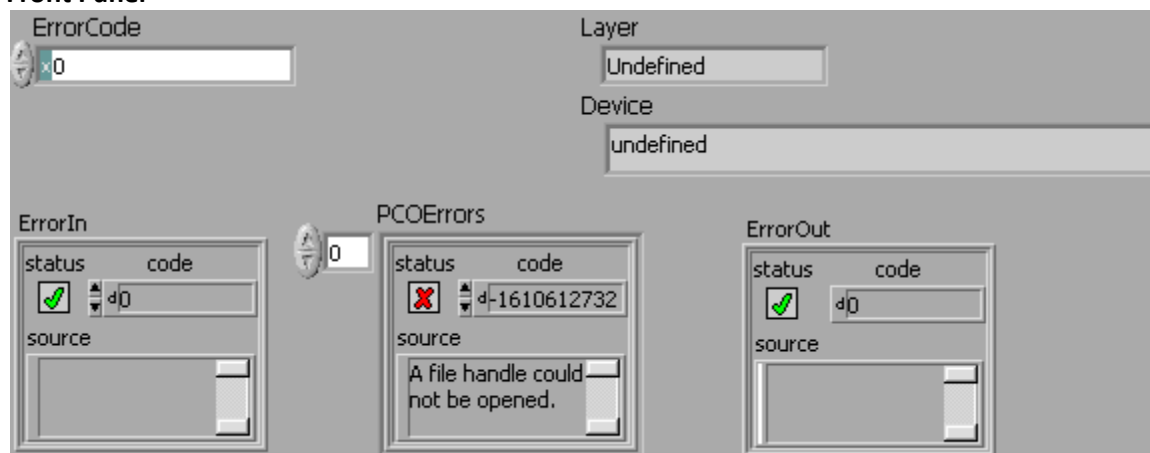
4.1.14. ErrorManager.vi

ErrorManager translates error numbers generated by the pco.camera interface into language. Information on the source of the error, the device and software layer where the error originated is coded into the error string. This function is called by all the camera interface functions, so that error numbers can be converted into LabVIEW error clusters for further handling.

Connector Pane



Front Panel



Controls and Indicators



PCOErrors List of possible error numbers and descriptions of the errors. Last element is reserved for unknown errors.




error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



ErrorCode Error code returned from any driver function



ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

 **ErrorOut** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

4.1.15. **FormatHWDESC.vi**

Connector Pane



Front Panel

Controls and Indicators

 **SC2_Hardware_DESC**

4.1.16. FormatSoftwareDesc.vi

Connector Pane



Front Panel

Controls and Indicators

 SC2_Software_DESC

 Cluster

 string

 MinorRevision

 MajorRevision

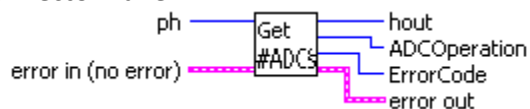
 Variant

4.2. Sensor.llb

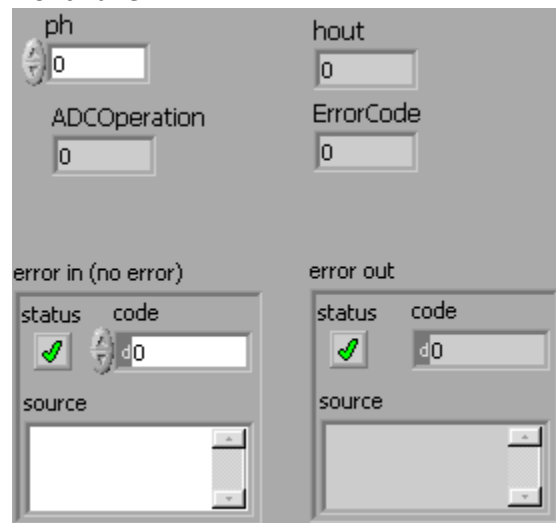
4.2.1. GetADCOperation.vi

Finds the number of A/D converters currently in use. Some models have multiple ADC's for faster readout.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode

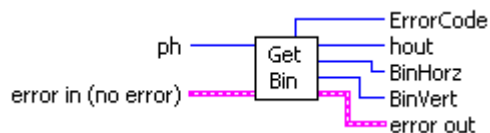


ADCOperation Number of A/D converters currently in use.

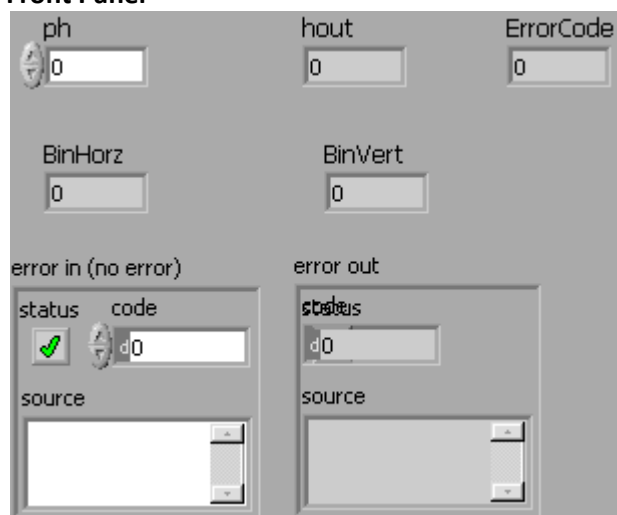
4.2.2. GetBinning.vi

Finds the camera's current binning setting as set by the Set Binning and ArmCamera commands. Use GetDescription.vi to determine what the allowed binning settings are for the camera.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



BinHorz Current horizontal binning setting

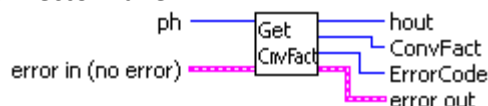


BinVert Current vertical binning setting

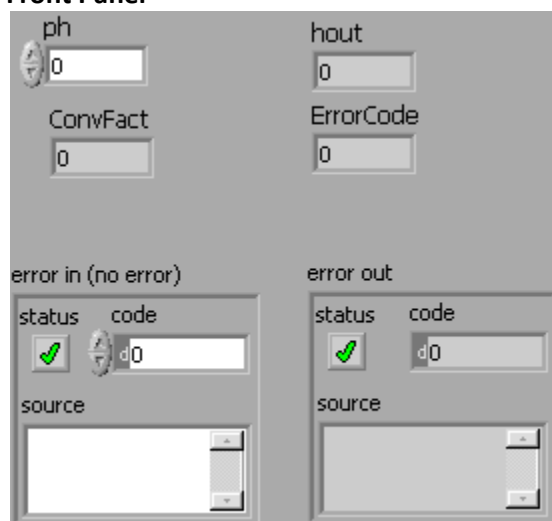
4.2.3. GetConversionFactor.vi

Finds the current A/D converter gain setting, in electrons/pixel. The number returned is an integer and represents 100 times the actual value, e.g. 435 = 4.35 electrons/count.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be

used by other VIs.



ErrorCode

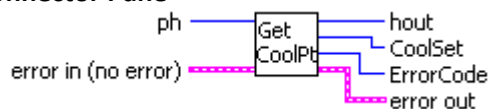


ConvFact Finds the current A/D converter gain setting, in electrons/pixel. The number returned is an integer and represents 100 times the actual value, e.g. 435 = 4.35 electrons/count.

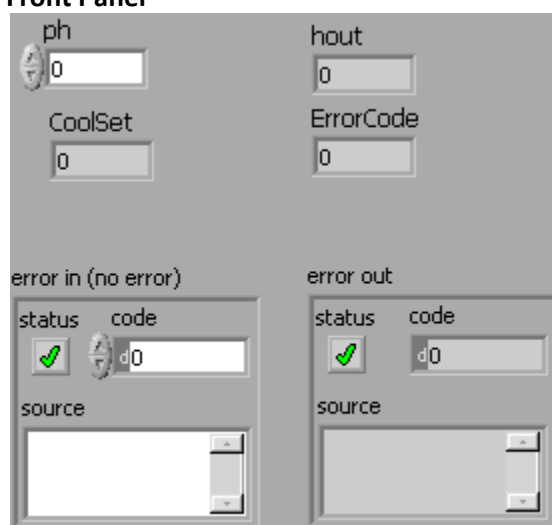
4.2.4. GetCoolingSetpointTemperature.vi

Finds the current cooling temperature setpoint, in °C. Value will be 0 for cameras which are not cooled.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



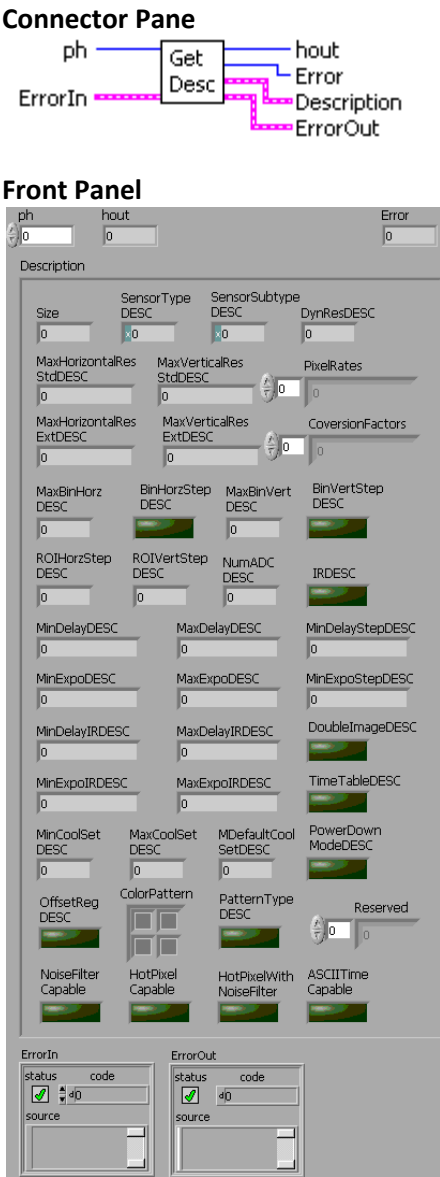
error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs

**ErrorCode**

CoolSet Current cooling temperature setpoint, in °C. Value will be 0 for cameras which are not cooled.

4.2.5. GetDescription.vi

Returns detailed information on the camera referenced by the input handle. The "Description" cluster contains information on sensor type, supported modes of operation, and various other operating parameters.



Controls and Indicators



ph Handle to camera



Description Detailed description of the properties of the camera including sensor type, digitization rate, etc.



Size Size of this cluster in bytes



SensorTypeDESC Identification code for image sensor in camera head

Here are the sensor codes defined for this release:

Monochrome sensors:

0x0010 Sony ICX285AL

0x0020 Sony ICX263AL

0x0030 Sony ICX274AL

0x0040 Sony ICX407AL

0x0050 Sony ICX414AL

0x0110 Kodak KAI-2000M

0x0120 Kodak KAI-2001M

0x0130 Kodak KAI-4010M

0x0140 Kodak KAI-4020M

0x0150 Kodak KAI-11000M

0x1010 Micron MV13 bw

Color sensors:

0x0011 Sony ICX285AK
0x0021 Sony ICX263AK
0x0031 Sony ICX274AK
0x0041 Sony ICX407AK
0x0051 Sony ICX414AK
0x0111 Kodak KAI-2000CM
0x0121 Kodak KAI-2001CM
0x0131 Kodak KAI-4010CM
0x0141 Kodak KAI-4020CM
0x0151 Kodak KAI-11000CM
0x1011 Micron MV13 col

U16 **SensorSubtypeDESC** Code for sensor subtypes, variants etc.

U16 **MaxHorizontalResStdDESC** Maximum horizontal effective pixels in standard mode

U16 **MaxVerticalResStdDESC** Maximum vertical effective pixels in standard mode

U16 **MaxHorizontalResExtDESC** Maximum horizontal pixels in extended mode. This includes all dark pixels, reference pixels and effective pixels

U16 **MaxVerticalResExtDESC** Maximum vertical pixels in extended mode. This includes all dark pixels, reference pixels and effective pixels

U16 **DynResDESC** Analog to digital converter resolution, in bits.

U16 **MaxBinHorzDESC** Maximum bin size in the horizontal direction

TF **BinHorzStepDESC** Describes the binning increments allowed in the horizontal direction.

FALSE = Only binary increments are allowed, i.e. 1, 2, 4, 8.....

TRUE = Linear increments are allowed, i.e. 1, 2, 3, 4, 5, 6,.....

U16 **MaxBinVertDESC** Maximum bin size in the vertical direction

TF **BinVertStepDESC** Describes the binning increments allowed in the vertical direction.

FALSE = Only binary increments are allowed, i.e. 1, 2, 4, 8.....

TRUE = Linear increments are allowed, i.e. 1, 2, 3, 4, 5, 6,.....

U16 **ROIHorzStepDESC** Describes the minimum increment allowed in the region of interest setting in the horizontal direction. For example:

A value of 10 means the right ROI border can be 1, 11, 21, etc.

A value of 32 means the right ROI border can be 1, 33, 65, etc.

U16 **ROIVertStepDESC** Describes the minimum increment allowed in the region of interest setting in the vertical direction. For example:

A value of 10 means the top ROI border can be 1, 11, 21, etc.

A value of 32 means the top ROI border can be 1, 33, 65, etc.

U16 **NumADCDESC** Number of A/D converters available

TF **IRDESC** Indicates whether camera has enhanced infrared mode

FALSE - IR enhancement not supported

TRUE - Camera supports enhanced infrared mode

U32 **MinDelayDESC** Minimum delay setting in nanoseconds for standard mode

U32 **MaxDelayDESC** Maximum delay setting in milliseconds for standard mode

U32 **MinDelayStepDESC** Minimum delay step size in nanoseconds (all modes)

U32 **MinExpoDESC** Minimum exposure setting in nanoseconds for standard mode

U32 **MaxExpoDESC** Maximum exposure setting in milliseconds for standard mode

U32 **MinExpoStepDESC** Minimum exposure step size in nanoseconds (all modes)

U32 **MinDelayIRDESC** Minimum delay setting in nanoseconds for IR enhanced mode

U32 **MaxDelayIRDESC** Maximum delay setting in milliseconds for IR enhanced mode

U32 **MinExpoIRDESC** Minimum exposure setting in nanoseconds for IR enhanced mode

U32 **MaxExpoIRDESC** Maximum exposure setting in milliseconds for IR enhanced mode

TF **TimeTableDESC** Indicates camera's ability to use delay / exposure time tables

FALSE - Exposure time tables not supported

TRUE - Exposure time tables supported

TF **DoubleImageDESC** Indicates double image capability

FALSE - Double image mode not supported.

TRUE - Camera has double image capability.

I16 **MinCoolSetDESC** Minimum cooling setpoint, in °C. Value is 0 for uncooled cameras

I16 **MaxCoolSetDESC** Maximum cooling setpoint, in °C. Value is 0 for uncooled cameras

I16 **MDefaultCoolSetDESC** Default cooling setpoint, in °C. Value is 0 for uncooled cameras



PowerDownModeDESC Indicates whether the sensor can be powered down to reduce dark current

FALSE - Power down mode is not supported

TRUE - Power down mode is supported



OffsetRegDESC Indicates whether camera is capable of automatic offset regulation. Offset regulation is performed by sampling dark reference pixels to gauge the thermal drift in the sensor, then adjusting the offset voltage to compensate.

FALSE - Camera does not support automatic offset regulation

TRUE - Offset regulation is supported.



ColorPattern Describes the pattern of color filters used on the pixels of a color sensor. This number packs 4, 4 bit numbers describing each quadrant of a 2 X 2 pixel color pattern cell.



UpperLeftColor Color of the upper left pixel in the 2 X 2 color pattern

Possible values are:

0 - Monochrome, no color filter

1 - Red

2 - Green A

3 - Green B

4 - Blue

- 5 - Cyan
- 6 - Magenta
- 7 - Yellow



UpperRightColor Color of the upper right pixel in the 2 X 2 color pattern

Possible values are:

- 0 - Monochrome, no color filter
- 1 - Red
- 2 - Green A
- 3 - Green B
- 4 - Blue
- 5 - Cyan
- 6 - Magenta
- 7 - Yellow



LowerLeftColor Color of the lower left pixel in the 2 X 2 color pattern

Possible values are:

- 0 - Monochrome, no color filter
- 1 - Red
- 2 - Green A
- 3 - Green B

4 - Blue

5 - Cyan

6 - Magenta

7 - Yellow



LowerRightColor Color of the lower right pixel in the 2 X 2 color pattern

Possible values are:

0 - Monochrome, no color filter

1 - Red

2 - Green A

3 - Green B

4 - Blue

5 - Cyan

6 - Magenta

7 - Yellow



PatternTypeDESC indicates, for color sensors, whether sensor has an RGB or a CMY Bayer pattern. For monochrome sensors, the output is always FALSE.

FALSE - RGB pattern

TRUE - CMY pattern



PixelRates Analog to digital converter rates, in samples per second. For cameras that support multiple A/D rates, each element of the array will

contain the sampling rate for each supported mode.

[U32] PixelRateDesc1

[U16] CoverisionFactors Digitizer conversion factors, in electrons / count. For cameras that support multiple ranges, each element of the array will be filled with the corresponding conversion factor for that mode.

[U16] ConvFactDesc1

[U32] Reserved Reserved for future use

[U32] Reserved1

[TF] NoiseFilterCapable Indicates whether camera is capable of noise filtering

FALSE - Camera does not support noise filter

TRUE - Noise filter is supported.

[TF] HotPixelCapable Indicates whether camera is capable of automatic hot pixel correction. NOTE: Also check the "HotPixelWithNoiseFilter" indicator to determine if the noise filter is required for this mode of operation.

FALSE - Camera does not support hot pixel correction

TRUE - hot PixelCorrection is supported.



HotPixelWithNoiseFilter Indicates whether noise filter is required for hot pixel correction

FALSE - Noise filter is not required

TRUE - Noise filter is required



ASCIITimeCapable Indicates whether camera is capable of automatic offset regulation. Offset regulation is performed by sampling dark reference pixels to gauge the thermal drift in the sensor, then adjusting the offset voltage to compensate.

FALSE - Camera does not support automatic offset regulation

TRUE - Offset regulation is supported.



hout Handle returned by GetDescription



ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

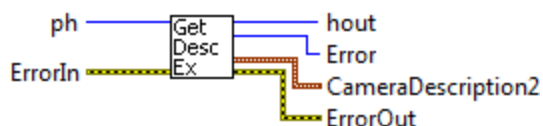


ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

4.2.6. GetDescriptionEX.vi

Returns more detailed information on camera operating parameters than is available from "GetDescription.vi". "Description2" contains information on modulation capability, etc.

Connector Pane



Front Panel

ph 0	hout 0	Error 0	
CameraDescription2			
MinPeriodicTime 0	MaxPeriodicTime 0	MinPeriodicCondition 0	MaxNumberExposures 0
MinMonitorSignalOffset 0	MaxMonitorSignalOffset 0	MinPeriodicalStep 0	StartTimeDelay 0
MinDelayMod 0	MaxDelayMod 0	MinDelayStep 0	MinMonitorStep 0
MinExposureMod 0	MaxExposureMod 0	MinExposureStep 0	ModulateCapabilities 0
Reserved 1 0	Reserved 2 0	Reserved 3 0	Reserved 4 0
Reserved 5 0	Reserved 6 0	Reserved 7 0	Reserved 8 0
Reserved 9 0	Reserved 10 0	Reserved 11 0	Reserved 12 0
Reserved 13 0	Reserved 14 0	Reserved 15 0	Reserved 16 0
ErrorIn		ErrorOut	
status code <input checked="" type="checkbox"/> 0 source		status code <input checked="" type="checkbox"/> 0 source	
Type Extended description			

Controls and Indicators



ph Handle to camera.



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.



Error Error code returned by driver.



hout Handle to camera.



ErrorOut The **error in** cluster can accept error information wired from VIs previously called.



CameraDescription2 Additional information about the cameras capabilities



MinPeriodicTime Smallest length of time allowed between exposures, in nanoseconds



MaxPeriodicTime Maximum length of time between exposures, in milliseconds



MinPeriodicCondition Smallest residual time within one period, in nanoseconds. The residual time is the difference between the period and the combination of delay and exposure within the period:

$$\text{residual} = \text{period} - (\text{delay} + \text{exposure})$$



MaxNumberExposures Maximum number of exposures within one frame time



MinMonitorSignalOffset Delay between offset signal and start of exposure, in nanoseconds. Negative value indicates that the monitor signal can be generated before the exposure starts.



MaxMonitorSignalOffset Maximum delay between start of exposure and offset signal, in nanoseconds.



MinPeriodicalStep Minimum step size for setting the periodic time, in nanoseconds. Period must be a multiple of this value.



StartTimeDelay Time delay between start of frame and start of modulation, in nanoseconds



MinMonitorStep Minimum step size for setting the monitor signal time, in nanoseconds. Monitor signal must be a multiple of this value



MinDelayMod Minimum delay between start of period and start of exposure, in nanoseconds



MaxDelayMod Maximum delay between start of period and start of exposure, in nanoseconds



MinDelayStep Minimum step size for setting the modulation delay time, in nanoseconds

nanoseconds. Delay must be a multiple of this value.



MinExposureMod Minimum exposure time in modulation mode (nanoseconds)



MaxExposureMod Maximum exposure time in modulation mode (nanoseconds)



MinExposureStep Minimum step size for setting the modulation exposure time, in nanoseconds. Exposure time must be a multiple of this value.



ModulateCapabilities Describes the modulation modes available for this camera.

Bit 0 - Modulation available if set.



Reserved 1 - 16 Reserved for future capability information

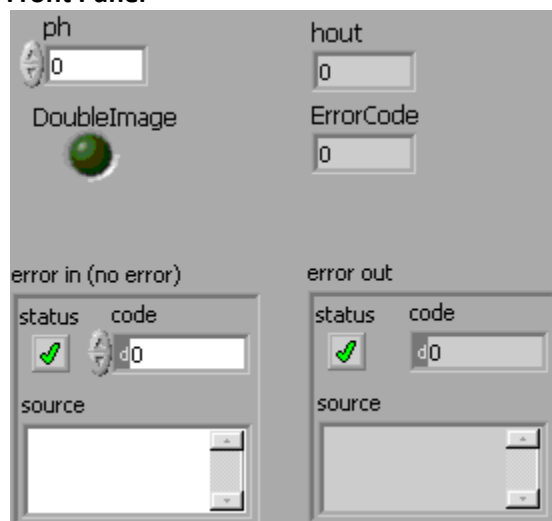
4.2.7. GetDoubleImageMode.vi

Determines if the camera is currently in double image mode.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

**ErrorCode****DoubleImage** Indicates the state of the double image mode.

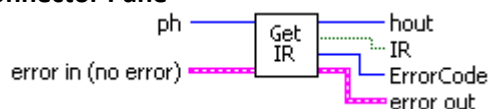
FALSE - Double image mode disabled

TRUE - Camera is currently in double image mode.

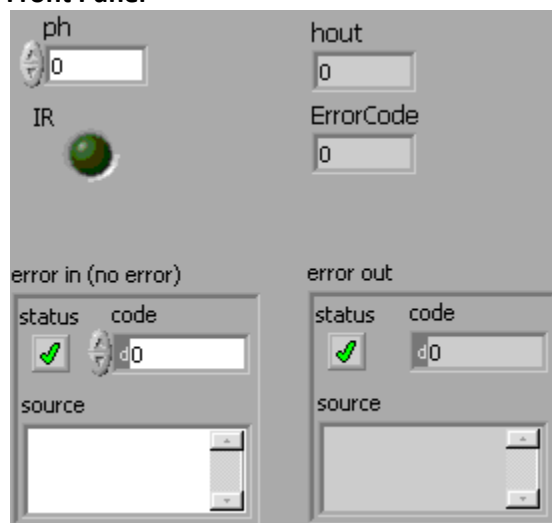
4.2.8. GetIRSensitivity.vi

Indicates the status of the enhanced infrared sensitivity mode.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

U32 ErrorCode

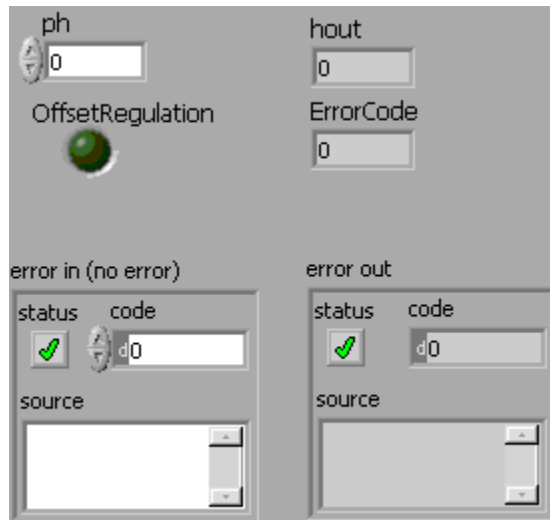
TF IR Indicates the status of the enhanced infrared sensitivity mode.

FALSE - Enhanced infrared sensitivity mode is disabled

FALSE - Enhanced infrared sensitivity mode is enabled

4.2.9. GetOffsetMode.vi

Returns the current state of the automatic offset regulation.

Connector Pane**Front Panel****Controls and Indicators**

U64 **ph** Handle for the camera

ETI **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed

in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



OffsetRegulation Current state of the automatic offset regulation.

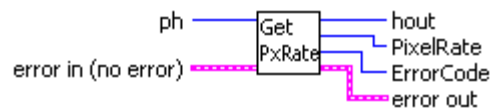
FALSE - Automatic offset control enabled

TRUE - Automatic offset control disabled

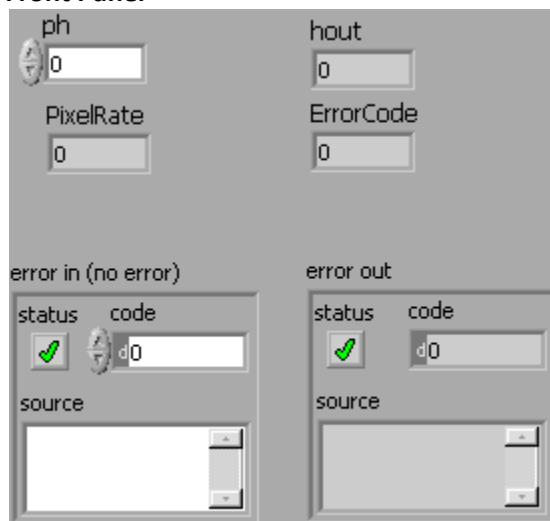
4.2.10. GetPixelRate.vi

Finds the current pixel rate, as of the last ArmCamera command. The pixelrate is given in Hz, e.g. 10000000 = 10 MHz

Connector Pane




Front Panel




Controls and Indicators

 **ph** Handle for the camera

 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

 **hout** Handle output

 **error out** The **error out** cluster passes error or warning information out of a VI to be

used by other VIs.

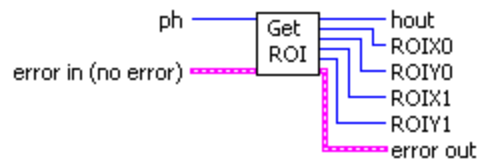
I32 ErrorCode

U32 PixelRate Current pixel rate, as of the last ArmCamera command. The pixelrate is given in Hz, e.g. 10000000 = 10 MHz

4.2.11. GetROI.vi

Get ROI (region or area of interest) window settings. The ROI is equal to or smaller than the absolute image area which is defined by the settings of format and binning.

Connector Pane



Front Panel

The front panel of the GetROI.vi block contains the following controls:

- Inputs:**
 - `ph`: A numeric control with a range of 0 to 10.
 - `ROI X0`: A numeric control with a range of 0 to 10.
 - `ROI X1`: A numeric control with a range of 0 to 10.
 - `ROI Y0`: A numeric control with a range of 0 to 10.
 - `ROI Y1`: A numeric control with a range of 0 to 10.
 - `error in (no error)`: A status indicator with a green checkmark and a code field set to 0.
 - `source`: A text area for error source details.
- Outputs:**
 - `hout`: A numeric control with a range of 0 to 10.
 - `ROI X0`: A numeric control with a range of 0 to 10.
 - `ROI Y0`: A numeric control with a range of 0 to 10.
 - `ROI X1`: A numeric control with a range of 0 to 10.
 - `ROI Y1`: A numeric control with a range of 0 to 10.
 - `error out`: A status indicator with a green checkmark and a code field set to 0.
 - `source`: A text area for error source details.

Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



ROIX0 Left border of the region of interest



ROIY0 Top border of the region of interest



ROIX1 Right border of the region of interest



ROIY1 Bottom border of the region of interest

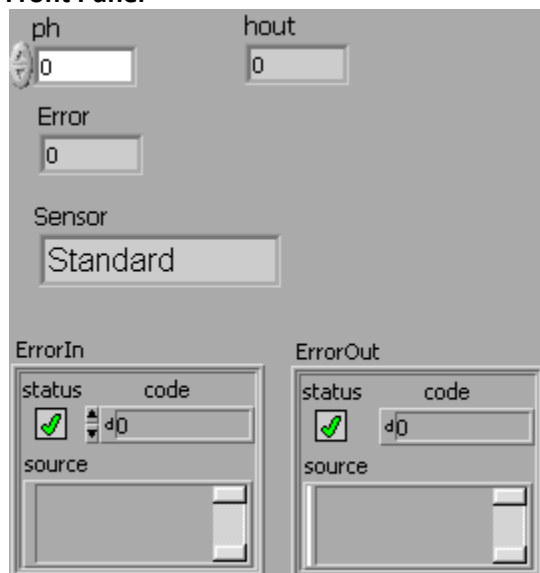
4.2.12. GetSensorFormat.vi

Returns the format of the sensor as either Standard or Extended. Extended format displays all pixels, including dark reference and dummies.

Connector Pane



Front Panel



Controls and Indicators



ph Handle to the camera



ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Error Error number returned by driver



hout Handle of the connected camera



Sensor Current sensor format setting:

0 - Standard format displays only active pixels

1 - Extended format: displays active dark reference and dummy pixels.

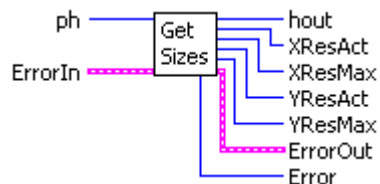


ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

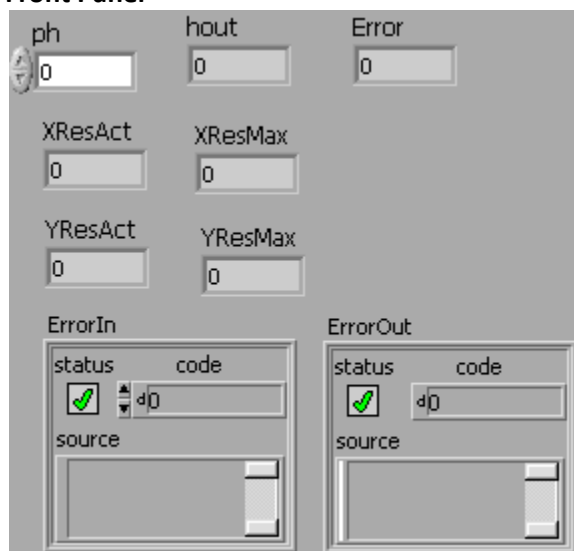
4.2.13. GetSizes.vi

Finds the actual size of the image as set by the last ArmCamera command.

Connector Pane



Front Panel



Controls and Indicators

U641 **ph** Handle for camera

E11 **ErrorIn** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

I32 **Error**



hout Handle of the connected camera



XResAct Actual horizontal resolution of the image as determined by the settings at the last ArmCamera command.



YResAct Actual vertical resolution of the image as determined by the settings at the last ArmCamera command.



XResMax Maximum horizontal resolution for this sensor



YResMax Maximum vertical resolution for this sensor



ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

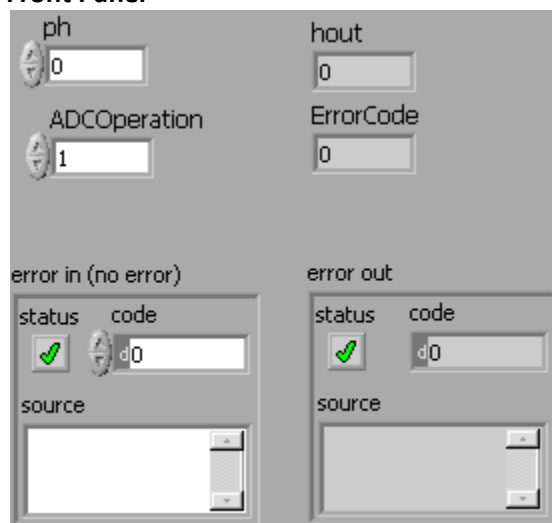
4.2.14. SetADCOperation.vi

Sets the number of A/D converters used to read out the image sensor. One ADC gives the highest linearity, but multiple ADC's can be used in some models for faster readout. Use GetDescription.vi to find the maximum number of ADC's available.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



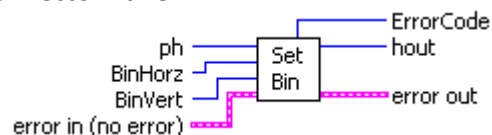
ADCOperation Sets the number of A/D converters used to read out the image sensor. One ADC gives the highest linearity, but multiple ADC's can be used in some models for faster readout. Use GetDescription.vi to find the maximum number of ADC's available.

**hout** Handle output**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.**ErrorCode**

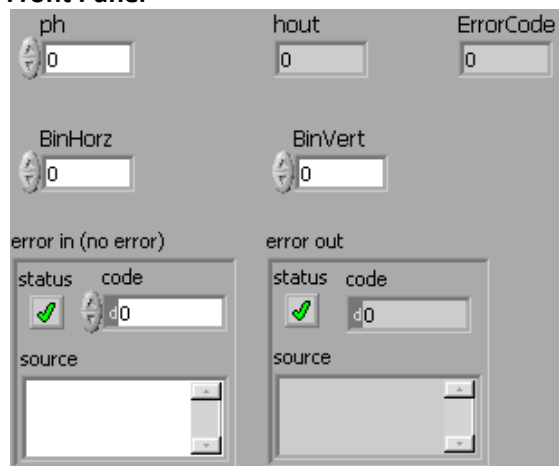
4.2.15. SetBinning.vi

Sets the binning to be used by the camera at the next ArmCamera command. Use GetDescription.vi to determine what the allowed binning settings are for the camera.

Connector Pane



Front Panel



Controls and Indicators

**ph** Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



BinHorz Set horizontal binning.



BinVert Set vertical binning.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

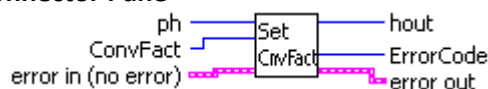


ErrorCode

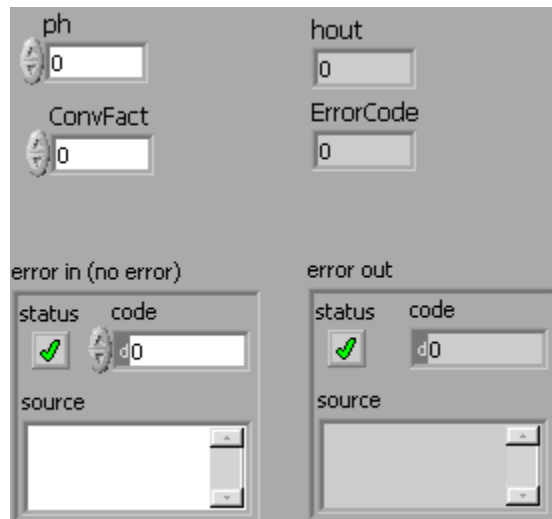
4.2.16. SetConversionFactor.vi

Sets the A/D converter gain setting, in electrons/pixel. This setting is an integer and represents 100 times the actual value, e.g. 435 = 4.35 electrons/count. Use GetDescription to determine the valid settings for the camera. Setting will take effect at next ArmCamera command.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



ConvFact Sets the A/D converter gain, in electrons/pixel. This setting is an integer and represents 100 times the actual value, e.g. 435 = 4.35 electrons/count. Use GetDescription to determine the valid settings for the camera. Setting will take effect at next ArmCamera command.

U64 **hout** Handle output

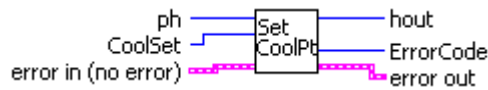
E31 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

I32 **ErrorCode**

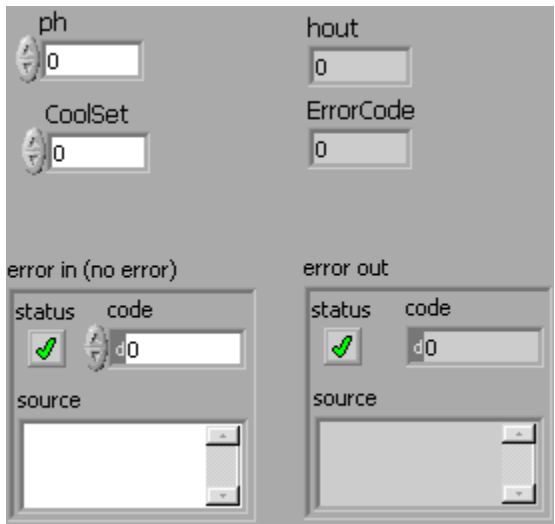
4.2.17. **SetCoolingSetpointTemperature.vi**

Sets the cooling temperature setpoint on cooled cameras. Use GetDescription.vi to determine if the camera supports this feature.

Connector Pane








Front Panel



Controls and Indicators

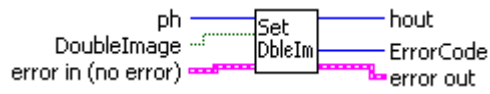
U64 **ph** Handle for the camera

- **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- **CoolSet** Desired cooling setpoint, in °C.
- **hout** Handle output
- **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
- **ErrorCode**

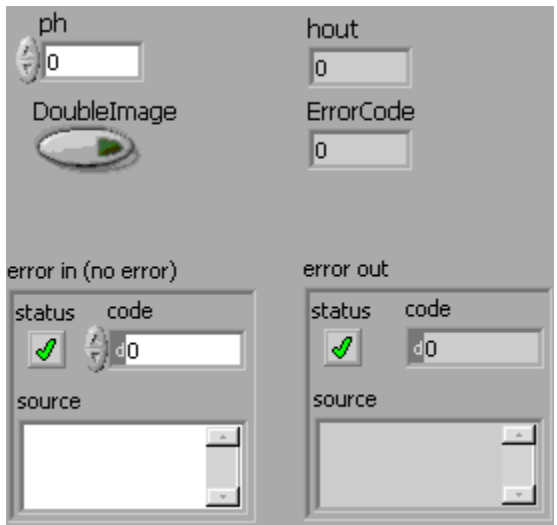
4.2.18. **SetDoubleImageMode.vi**

Enables or disables double image mode in cameras that have this capability. Use GetDescription.vi to determine if the camera has double image capability. This setting will take effect at the next ArmCamera command.


Connector Pane





Front Panel



Controls and Indicators

 **ph** Handle for the camera


 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

 **DoubleImage** Enables or disables double image mode in cameras that have this capability. Use GetDescription.vi to determine if the camera has double image capability. This setting will take effect at the next ArmCamera command.

FALSE - Disable double image mode

TRUE - Enable double image mode.

 **hout** Handle output

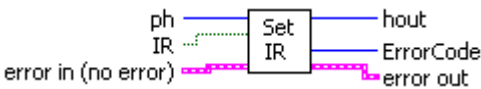
 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

 **ErrorCode**

4.2.19. SetIRSensitivity.vi

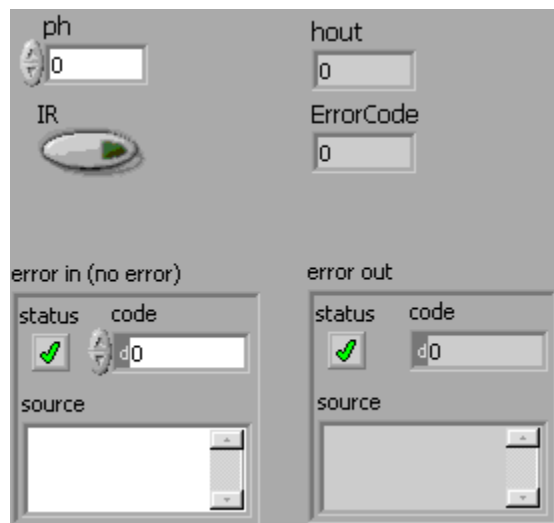
Enables or disables the enhanced infrared sensitivity mode, in cameras where this feature is supported. Use GetDescription.vi to determine if the camera supports this mode.

Connector Pane



Front Panel

File:	Version:	as of:	Author:	Page 106 of 252
MA_DCLABVIEWver204.doc	2.04	03.09.2015	MM/FRE/ LWA/ EO/ GHO	



Controls and Indicators

U64 **ph** Handle for the camera

E4 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

TF **IR** Enables or disables the enhanced infrared sensitivity mode, in cameras where this feature is supported. Use GetDescription.vi to determine if the camera supports this mode.

FALSE - Disable the enhanced IR mode

TRUE - Enable the enhanced IR mode

U64 **hout** Handle output

E4 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

I32 **ErrorCode**

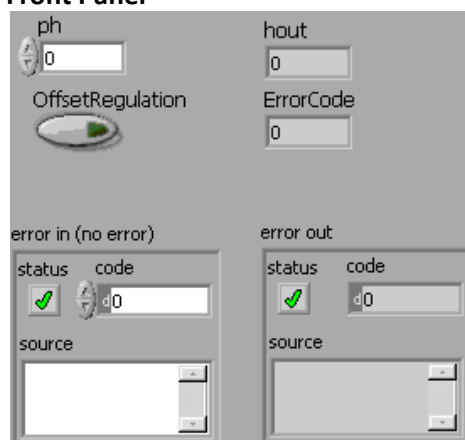
4.2.20. SetOffsetMode.vi

Enables or disables automatic offset regulation. Automatic offset regulation adjusts the offset based on measurement of the reference pixels.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



OffsetRegulation Enables or disables automatic offset regulation.

FALSE - Enable automatic offset regulation.

TRUE - Disable Automatic offset regulation.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

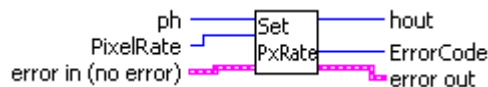


ErrorCode

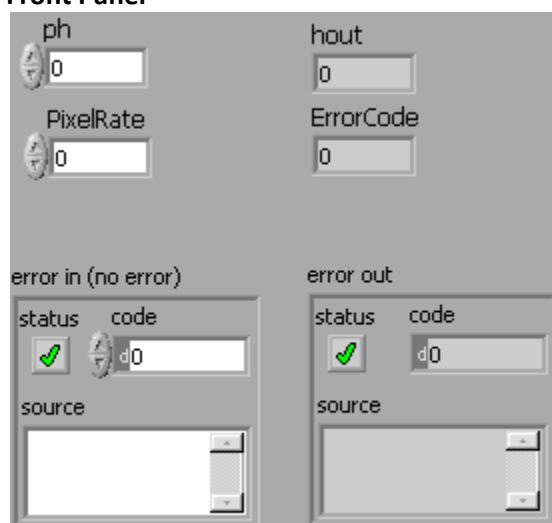
4.2.21. SetPixelRate.vi

Sets the pixel rate. Takes effect at next ArmCamera command. The pixelrate is specified in Hz, e.g. 10000000 = 10 Mhz. Use GetDescription.vi to find the allowed rates for the camera.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera




error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



PixelRate Sets the pixel rate. Takes effect at next ArmCamera command. The pixelrate is specified in Hz, e.g. 10000000 = 10 Mhz. Use GetDescription.vi to find the allowed rates for the camera.



hout Handle output

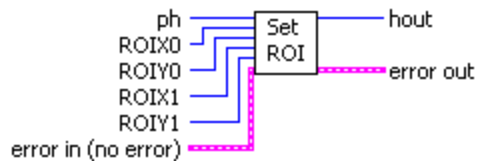
 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

 **ErrorCode**

4.2.22. SetROI.vi

Set ROI (region or area of interest) window. The ROI must be equal to or smaller than the absolute image area which is defined by the settings of format and binning. If you change the binning settings you have to adapt the ROI, before you call ArmCamera.vi. The binning setting sets the limits for the ROI. E.g. a sensor with 1600x1200 and binning 2x2 will result in a maximum ROI of 800x600.

Connector Pane



Front Panel



Controls and Indicators

- U64

ph Handle for the camera
- Err

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- U16

ROIX0 Sets left border of the region of interest
- U16

ROIY0 Sets top border of the region of interest
- U16

ROIX1 Sets right border of the region of interest
- U16

ROIY1 Sets bottom border of the region of interest
- U64

hout Handle output
- Err

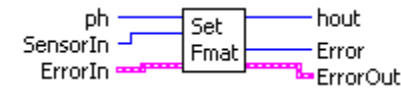
error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
- I32

ErrorCode

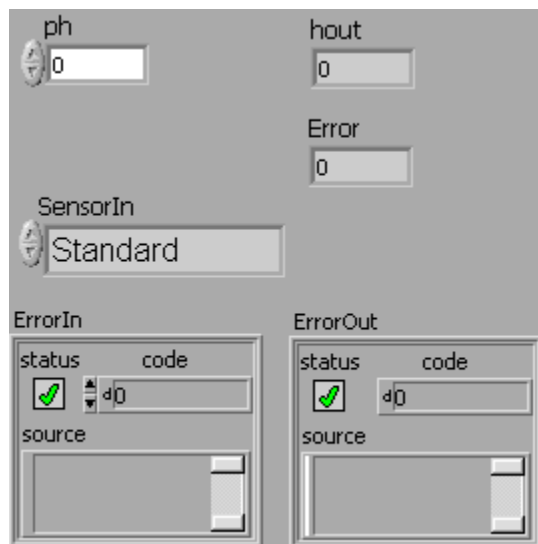
4.2.23. SetSensorFormat.vi

Sets the format of the sensor to either Standard or Extended. Extended format displays all pixels, including dark reference and dummies.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



SensorIn Sets sensor format:

0 - Standard format displays only active pixels

1 - Extended format: displays active dark reference and dummy pixels.



ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

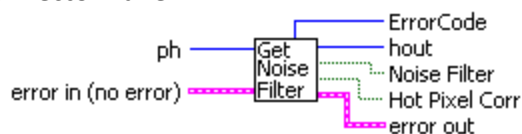


ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

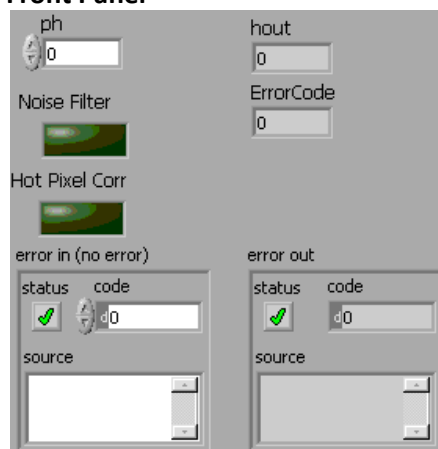
4.2.24. GetNoiseFilterMode.vi

Determines the current state of automatic noise filtering and hot pixel correction in camera models equipped with this feature.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



Noise Filter Indicates state of noise filter mode

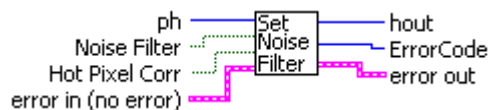


Hot Pixel Corr Indicates state of hot pixel correction

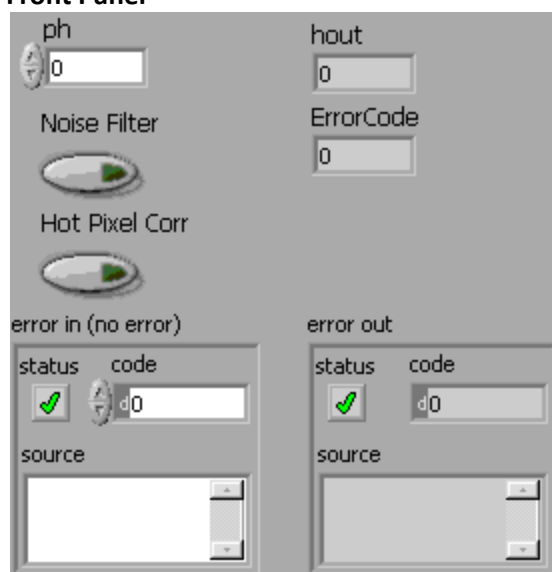
4.2.25. SetNoiseFilterMode.vi

Enables automatic noise filtering and hot pixel correction in camera models equipped with this feature.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera




error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.




Noise Filter Turn on noise filter mode



Hot Pixel Corr Enable hot pixel correction

- 

hout Handle output
- 

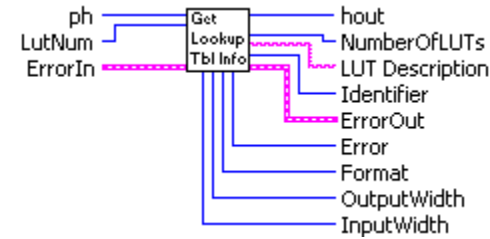
error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
- 

ErrorCode

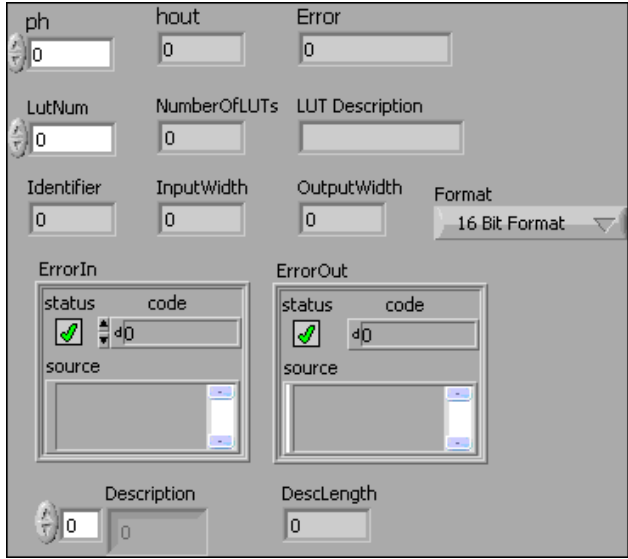
4.2.26. **GetLookupTableInfo.vi**

Gets information on lookup tables.

Connector Pane



Front Panel



Controls and Indicators



ph Camera handle input



ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



LutNum Number of the lookup table to query.



Error



hout Camera handle output.



ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



NumberOfLUTs Number of lookup tables defined.



Description Describes the lookup table contents. Should give some indication of how the table was constructed.



DescLength Length of the descriptor string.



Identifier Defines the unique identifier of the selected lookup table. Use this identifier in SetLookupTable.



InputWidth Number of bits on the LUT input.



OutputWidth Number of bits on the LUT output.



Format Format of the lookup table values.

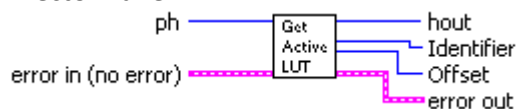


LUT Description Text description of the lookup table function. For example "sqrt(256*x)" describes how the input number is converted to the output value by multiplying by 256 and then taking the square root.

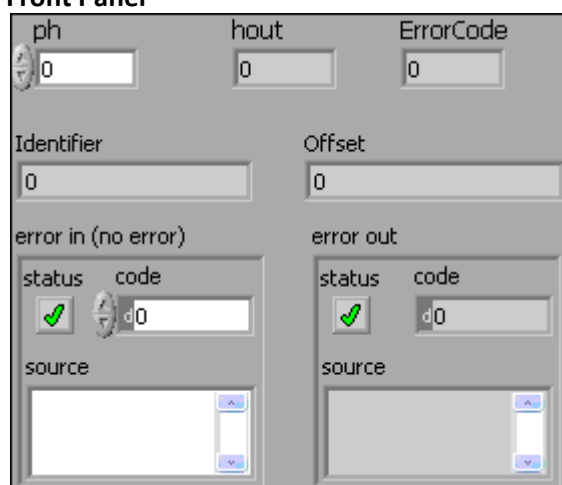
4.2.27. GetActiveLookupTable.vi

Determines which LUT is active.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



Identifier Unique ID of the active LUT. 0 if no LUT is active. Use Get LookupTableInfo.vi to find the codes for the lookup tables.

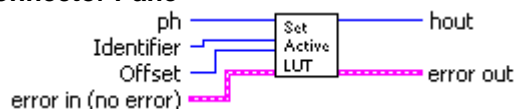


Offset 11 bit value for fixed offset subtraction before applying the lookup table.

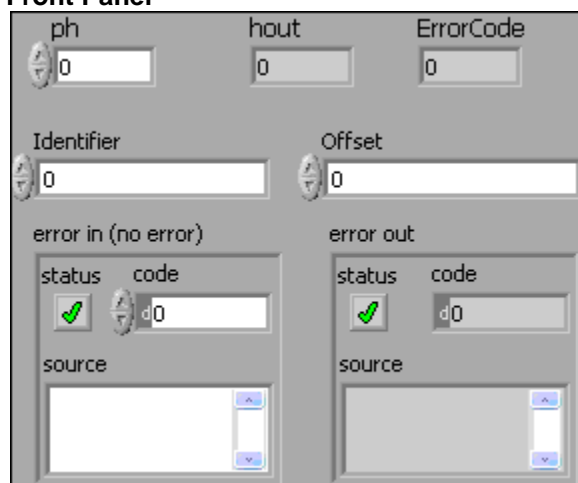
4.2.28. SetActiveLookupTable.vi

Sets the active LUT, or disables all LUTs

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Identifier Sets the LUT using predefined identifier. Set to 0 to disable LUTs. Use Get LookupTableInfo.vi to find the codes for the lookup tables.



Offset 11 bit value for fixed offset subtraction before applying the lookup table.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

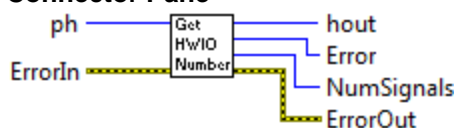


ErrorCode

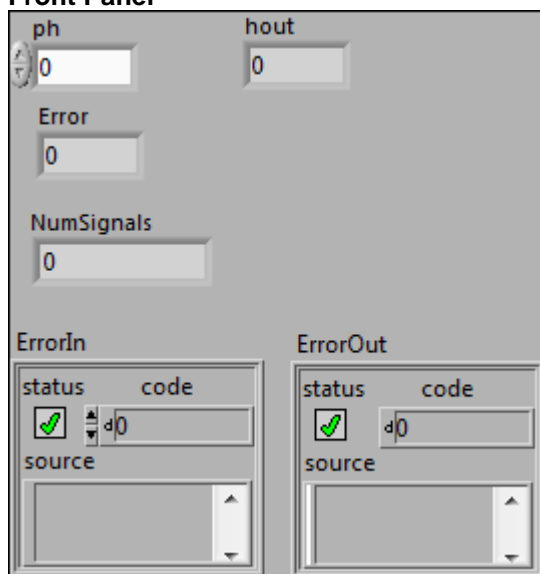
4.2.29. GetHWIOSignalCount.vi

Returns the number of Hardware I/O signals available..

Connector Pane



Front Panel



Controls and Indicators



ph Handle to the camera



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.



Error error number from driver



hout Handle to camera



NumSignals Number of Hardware Input/Output signals available for use.



ErrorOut The **error in** cluster can accept error information wired from VIs previously called

4.2.30. GetHWIOSignalDescriptor.vi


Provides details of the selected hardware signal, including signal names and I/O modes


Connector Pane













Front Panel

Controls and Indicators

 **ph** Handle to an open camera connection

 **ErrorIn** The **error in** cluster can accept error information wired from VIs previously called.

 **SignalNumber** Number of the HWIO signal to interrogate.

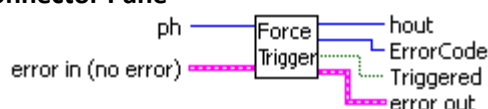
-  **Error** Error code returned by driver.
-  **hout** Handle to an open camera connection (same as input handle)
-  **ErrorOut** The **error in** cluster can accept error information wired from VIs previously called.
-  **SingleSignalDesc** Information on the selected I/O signal
-  **Names** Name given to this I/O signal, along with any alternates
-  **SignalDefinitions** Flags for signal options. Option is available if bit is 1 (TRUE). Unlisted bits are reserved for future use.
- Bit0: Signal can be enabled/disabled
Bit 1: Signal is status (output)
- Other bits reserved for future use
-  **SignalTypes** Flags for the selectable I/O options. Option is supported if bit is 1 (TRUE). Unlisted bits are reserved for future use
- Bit0: TTL
Bit1: Higher than TTL voltage
Bit2: Contact closure
Bit3: RS485 differential signaling
-  **SignalPolarity** Flags for level or transition triggering of I/O signal. Option is available if bit is 1 (TRUE). Unlisted bits are reserved for future use.
- Bit0: Low level active
Bit 1: High level active
Bit 2: Rising edge active
Bit 3: Falling edge active
-  **SignalFilter** Flags for signal filter options. Option is available if bit is 1 (TRUE). Unlisted bits are reserved for future use.
- Bit0: Filter can be disabled.
Bit1: Medium-level filter availability (t > 1 us)
Bit 2: High level filter available (t > 100 ms)

4.3. TimingControl.llb

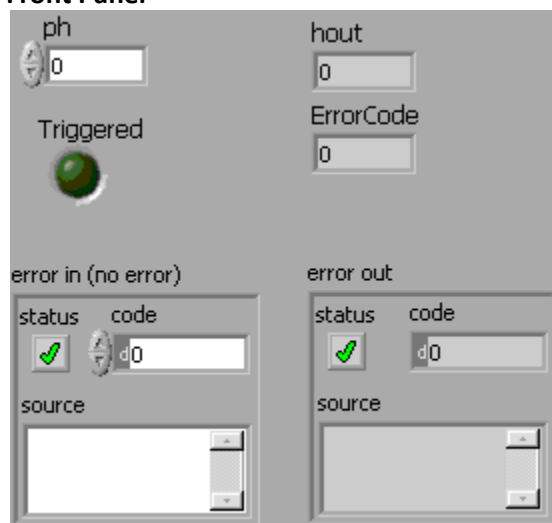
4.3.1. ForceTrigger.vi

This software command starts an exposure if the trigger mode is in the state [software trigger] (0x0001) or in the state [extern exposure & software trigger] (0x0002). If in state [extern exposure control] (0x0003), nothing happens. The camera has to be ready: (recording = [start]) and [not busy].

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

**ErrorCode****Triggered** Indicates if the trigger was successful.

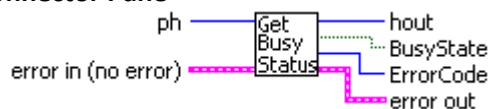
FALSE = trigger command was not successful: camera is busy

TRUE = a new image exposure has been triggered by the command

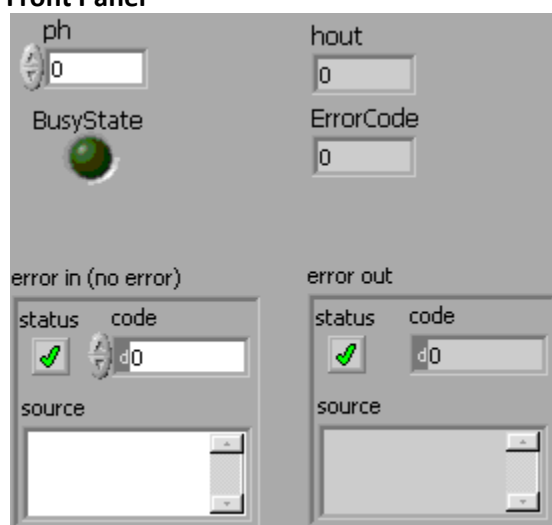
4.3.2. GetBusyStatus.vi

Finds the busy status of the camera. A camera is busy if it is exposing or if the sensor is being read out

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode

TF **BusyState** Busy status of the camera.

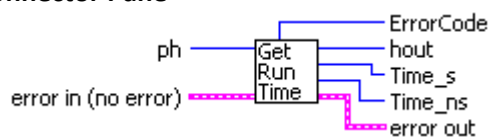
FALSE - Camera is not busy and can accept triggers

TRUE - Camera is busy and cannot accept triggers

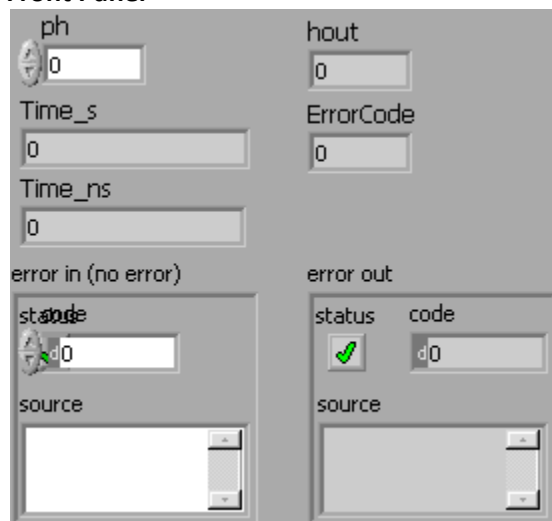
4.3.3. GetCOCRunTime.vi

Returns the time to execute the camera operation code, including all delay and exposure. This can be used to calculate the frames per second.

Connector Pane



Front Panel



Controls and Indicators

U64I **ph** Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



Time_s Number of seconds to execute the COC. Total time also includes the "Time_ns".

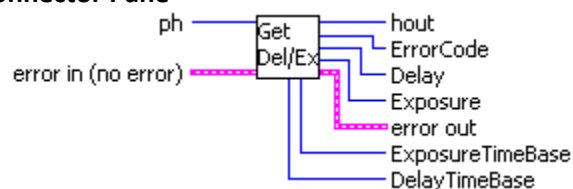


Time_ns Number of nanoseconds to execute the COC. Total time also includes the "Time_s".

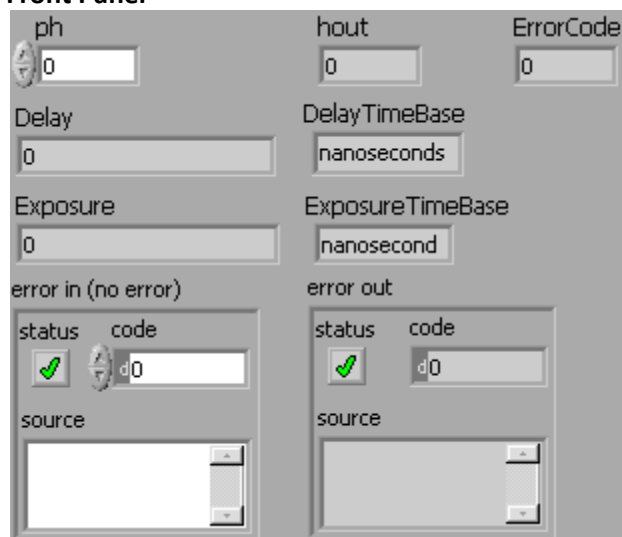
4.3.4. GetDelayExposureTime.vi

Returns the current delay and exposure time values.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

I32 **ErrorCode****U32** **Delay** Current delay settings, in timebase units.**U32** **Exposure** Current exposure settings, in timebase units.**U16** **DelayTimeBase** Time base (units of time) for the current delay setting.

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds

U16 **ExposureTimeBase** Time base (units of time) for the current exposure setting.

0 - Nanoseconds

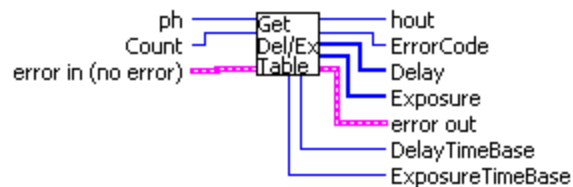
1 - Microseconds

2 - Milliseconds

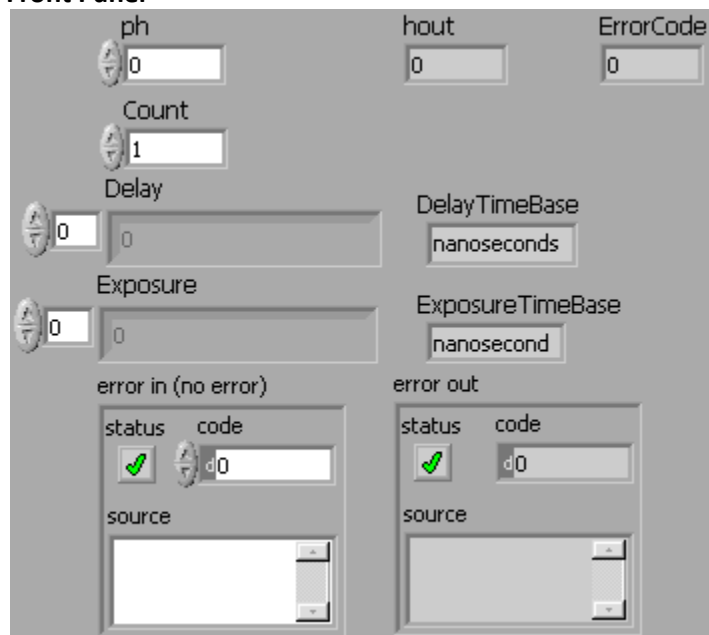
4.3.5. GetDelayExposureTimeTable.vi

Get delay / exposure time table. For some camera types it is possible to define a table with delay / exposure times (defined in the camera description). After start of exposure the camera will take a series of consecutive images with delay and exposure times as defined in the table. Therefore a flexible message format has been defined. The table consists of maximum 16 delay / exposure time pairs. If an exposure time entry is set to the value zero, then at execution time this delay/ exposure pair is disregarded and the sequence is started automatically with the first entry in the table. This results in a sequence of 1 to 16 images with different delay and exposure time settings. External or automatic triggering of images is fully functional for every image in the sequence. If the user wants maximum speed (at CCDs overlapping exposure and read out is taken), [auto trigger] should be selected and the sequence should be controlled with the <acq enbl> input.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Count Number of delay/exposure pairs defined in the table. Maximum number is 16 pairs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



Delay Array of delay values, in time base units



Exposure Array of exposure values, in time base units



ExposureTimeBase Time base (units of time) for the exposure setting. One time base is used for all exposure settings in the array

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds



DelayTimeBase Time base (units of time) for the delay setting. One time base is used for all delay settings in the array

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds

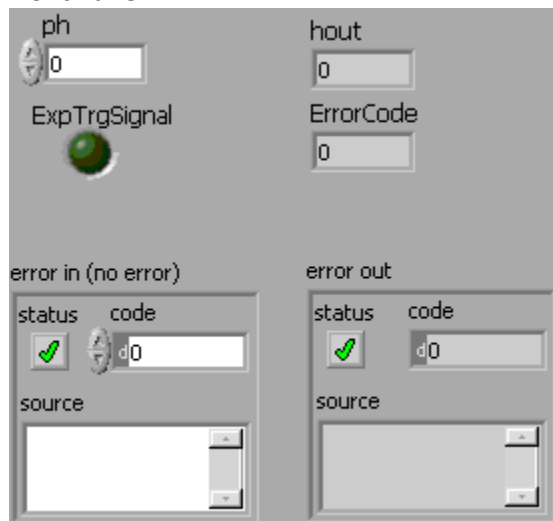
4.3.6. GetExpTrigSignalStatus.vi

Get the current status of the <exp trig> user input (one of the <control in> inputs). If the signal level at the <exp trig> input is HIGH and the DIP switch is set to HIGH, then the Status is TRUE. If the signal level at the <exp trig> input is HIGH and the DIP switch is set to LOW then the Status is FALSE.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

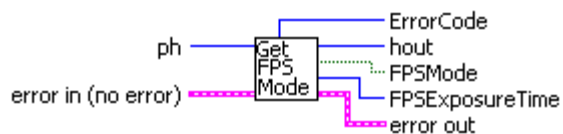
**hout** Handle output**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.**ErrorCode****ExpTrgSignal** Current status of the <exposure in> signal

4.3.7. GetFPSExposureMode.vi

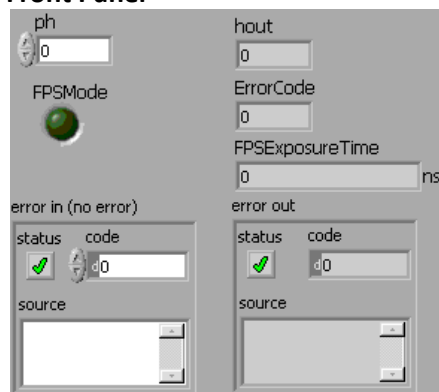
Determines if the camera is in frames per second (FPS) mode (available for the pco.1200hs camera model only!)

The FPS exposure mode is useful if you want to get the maximum exposure time for maximum frame rate. The maximum image framerate depends on pixelrate, vertical ROI and exposure time.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



FPSMode Status of the FPS mode

FALSE - FPS mode is not enabled

TRUE - FPS mode is enabled

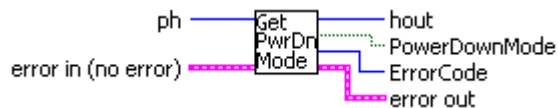


FPSExposureTime The exposure time that will be set if "FPS Exposure Mode" is on. The exposure time depends on the current settings of vertical ROI and Pixelrate. The returned time is always in ns!

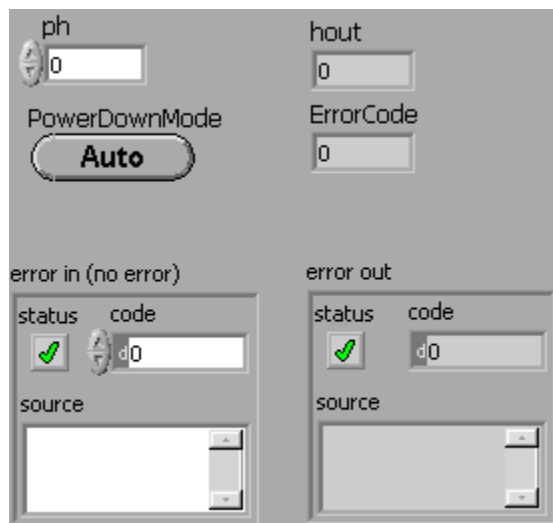
4.3.8. GetPowerDownMode.vi

Determines the current power down mode.

Connector Pane



Front Panel



Controls and Indicators

U64 **ph** Handle for the camera

E7 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

U64 **hout** Handle output

E7 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

I32 **ErrorCode**

TF **PowerDownMode** Current power down mode

FALSE - Power down is performed automatically

TRUE - Power down is under user control

4.3.9. GetTrigger.vi

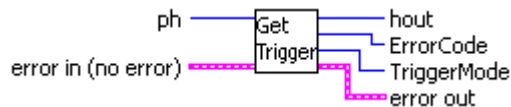
Find the current trigger mode of the camera.

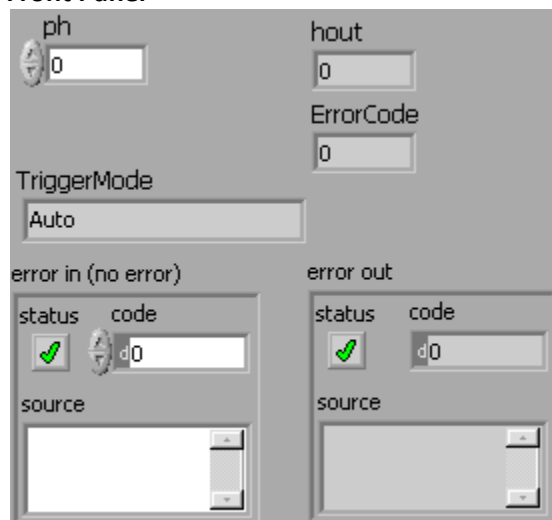
Trigger mode:

- 0x0000 = [auto trigger] An exposure of a new image is started automatically best possible compared to the readout of an image. If using a CCD and images are taken in a sequence, then exposures and readout of the sensor are started simultaneously. Signals at the trigger input (<exp trig>) are irrelevant.
- 0x0001 = [software trigger]: An exposure can only be started by a force trigger command.
- 0x0002 = [extern exposure & software trigger]: A delay / exposure sequence is started at the RISING or FALLING edge (depending on the DIP switch setting) of the trigger input (<exp trig>).
- 0x0003 = [extern exposure control]: The exposure time is defined by the pulse length at the trigger input(<exp trig>). The delay and exposure time values defined by the set/request delay and exposure command are ineffective. (Exposure time length control is also possible for double image mode; exposure time of the second image is given by the readout time of the first image.)

Note: Modes [extern exposure & software trigger] and [extern exposure control], depend also on the selected acquire mode. A trigger edge at the trigger input (<exp trig>) will be effective or not (see also SetAcquireMode.vi). A software trigger however will always be effective independent of the state of the <acq enbl> input.

Connector Pane



Front Panel**Controls and Indicators**

U64 **ph** Handle for the camera

ERR **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

U64 **hout** Handle output

ERR **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

I32 **ErrorCode**

U16 **TriggerMode** Current trigger mode of the camera.

Trigger mode:

- 0x0000 = [auto trigger] An exposure of a new image is started automatically best possible compared to the readout of an image. If using a CCD and images are taken in a sequence, then exposures and readout of the sensor are started simultaneously. Signals at the trigger input (<exptrig>) are irrelevant.

- 0x0001 = [software trigger]: An exposure can only be started by a force trigger

command.

- 0x0002 = [extern exposure & software trigger]: A delay / exposure sequence is started at the RISING or FALLING edge (depending on the DIP switch setting) of the trigger input (<exp trig>).

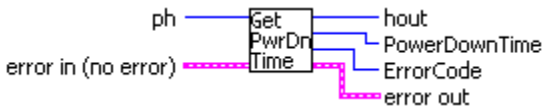
- 0x0003 = [extern exposure control]: The exposure time is defined by the pulse length at the trigger input(<exp trig>). The delay and exposure time values defined by the set/request delay and exposure command are ineffective. (Exposure time length control is also possible for double image mode; exposure time of the second image is given by the readout time of the first image.)

Note: Modes [extern exposure & software trigger] and [extern exposure control], depend also on the selected acquire mode. A trigger edge at the trigger input (<exp trig>) will be effective or not (see also SetAcquireMode.vi). A software trigger however will always be effective independent of the state of the <acq enbl> input.

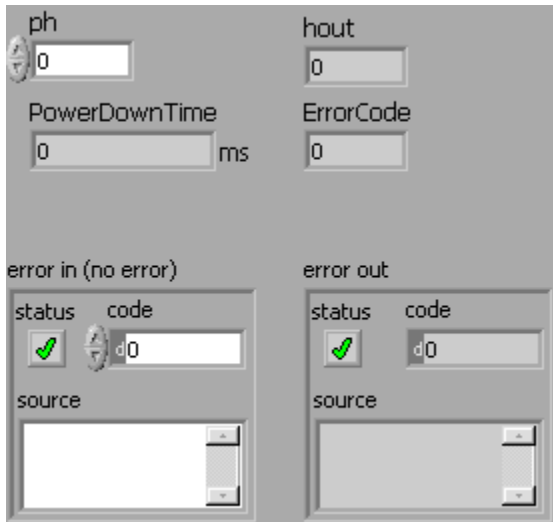
4.3.10. **GetUserPowerDownTime.vi**

Gets the current user power down time setting





Connector Pane



Front Panel



Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
-  **hout** Handle output
-  **error out** The **error out** cluster passes error or warning information out of a VI to be

used by other VIs.

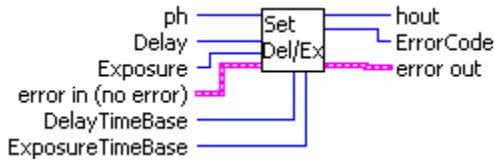
132 **ErrorCode**

U16 **PowerDownTime** The time to power down, in milliseconds, for user power down mode

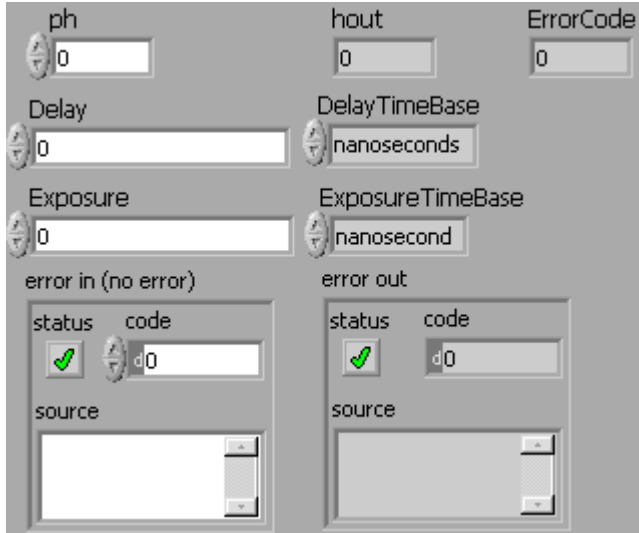
4.3.11. SetDelayExposureTime.vi

Sets delay and exposure time for the next exposure. Settings take effect after the next ArmCamera camera command. Use GetDescription.vi to determine the maximum and minimum settings for these parameters.

Connector Pane



Front Panel



Controls and Indicators

U64 **ph** Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Delay Delay, in timebase units. Use GetDescription.vi to determine the maximum and minimum values for the camera



Exposure Exposure, in timebase units. Use GetDescription.vi to determine the maximum and minimum values for the camera



DelayTimeBase Time base (units of time) for the delay setting.

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds



ExposureTimeBase Time base (units of time) for the exposure setting.

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



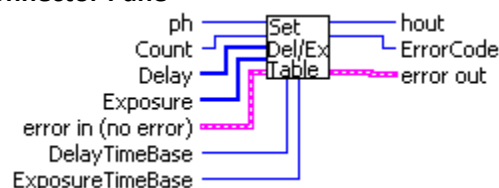
ErrorCode

4.3.12. SetDelayExposureTimeTable.vi

For some camera types it is possible to define a table with delay / exposure times. Use GetDescription.vi to determine if the camera supports this feature. After start of exposure the camera will take a series of consecutive images with delay and exposure times as defined in the table. Therefore a flexible message format has been defined. The table consists of maximum 16 delay / exposure time pairs. If an exposure time entry is set to the value zero, then at execution time this delay/ exposure pair is disregarded and the sequence is started automatically with the first entry in the table. This results in a sequence of 1 to 16 images with different delay and exposure time settings. External or automatic triggering of images is fully functional for every image in the sequence. If the user wants maximum speed (at CCDs overlapping exposure and read out is taken), [auto trigger] should be selected and the sequence should be controlled with the <acq enbl> input.

SetDelayExposureTime.vi and SetDelayExposureTimeTable.vi can only be used alternatively. Each of these functions will overwrite the settings of the other. Using SetDelayExposureTime.vi has the same effect as SetDelayExposureTimeTable.vi with the second exposure entry set to zero.

Connector Pane



Front Panel

Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Count Number of delay/exposure pairs defined in the table. Maximum number is 16 pairs



Delay Array of exposure values, in time base units.



Exposure Array of exposure values, in time base units



ExposureTimeBase Time base (units of time) for the exposure settings. One time base is used for all exposure settings in the array

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds



DelayTimeBase Time base (units of time) for the delay settings. One time base is used for all delay settings in the array

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds



hout Handle output



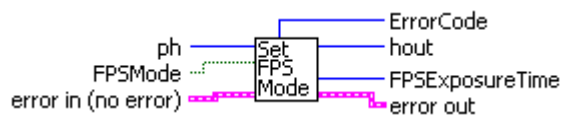
error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

4.3.13. SetFPSExposureMode.vi

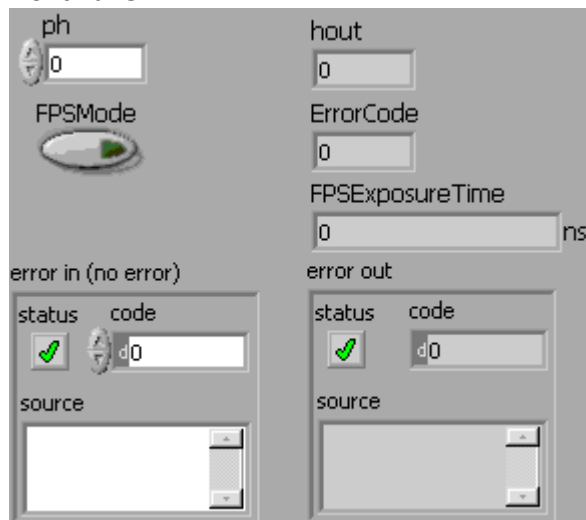
Enables or disables the frames per second (FPS) mode (available for the pco.1200hs camera model only!)

The FPS exposure mode is useful if you want to get the maximum exposure time for maximum frame rate. The maximum image framerate depends on pixelrate, vertical ROI and exposure time.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.




FPSMode Turns FPS mode on or off


FALSE - Disable FPS mode

TRUE - Enable FPS mode

 **hout** Handle output

 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

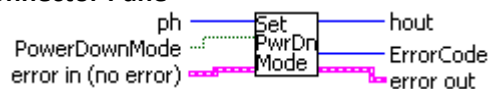
 **ErrorCode**

 **FPSExposureTime** The exposure time that will be set if “FPS Exposure Mode” is on. The exposure time depends on the current settings of vertical ROI and Pixelrate. The returned time is always in ns!

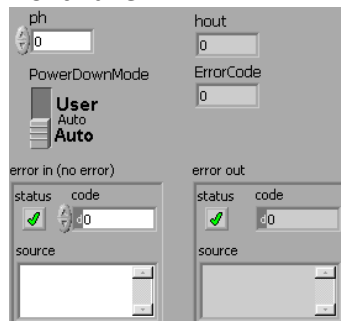
4.3.14. SetPowerDownMode.vi

Sets the power down mode of the camera. The camera powers down the output amplifier during long exposure times to reduce noise. In automatic mode, the power down is performed according to a preset firmware setting. In user mode, the power down is performed after a user-specified time. Use GetCameraDescription.vi to determine if the camera supports user power down mode.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



PowerDownMode Sets the power down mode

FALSE - Power down is performed automatically

TRUE - Power down is under user control



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode

4.3.15. SetTrigger.vi

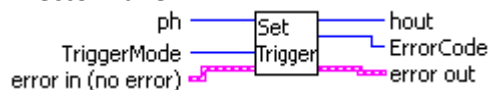
Set the current trigger mode of the camera.

Trigger mode:

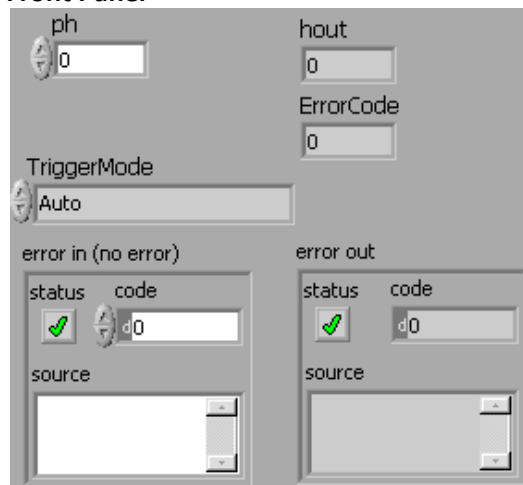
- 0x0000 = [auto trigger] An exposure of a new image is started automatically best possible compared to the readout of an image. If using a CCD and images are taken in a sequence, then exposures and readout of the sensor are started simultaneously. Signals at the trigger input (<exp trig>) are irrelevant.
- 0x0001 = [software trigger]: An exposure can only be started by a force trigger command.
- 0x0002 = [extern exposure & software trigger]: A delay / exposure sequence is started at the RISING or FALLING edge (depending on the DIP switch setting) of the trigger input (<exp trig>).
- 0x0003 = [extern exposure control]: The exposure time is defined by the pulse length at the trigger input(<exp trig>). The delay and exposure time values defined by the set/request delay and exposure command are ineffective. (Exposure time length control is also possible for double image mode; exposure time of the second image is given by the readout time of the first image.)

Note: Modes [extern exposure & software trigger] and [extern exposure control], depend also on the selected acquire mode. A trigger edge at the trigger input (<exp trig>) will be effective or not (see also SetAcquireMode.vi). A software trigger however will always be effective independent of the state of the <acq enbl> input.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



TriggerMode Sets the trigger mode of the camera.

Trigger mode:

- 0x0000 = [auto trigger] An exposure of a new image is started automatically best possible compared to the readout of an image. If using a CCD and images are taken in a sequence, then exposures and readout of the sensor are started simultaneously. Signals at the trigger input (<exp trig>) are irrelevant.

- 0x0001 = [software trigger]: An exposure can only be started by a force trigger command.

- 0x0002 = [extern exposure & software trigger]: A delay / exposure sequence is started at the RISING or FALLING edge (depending on the DIP switch setting) of the trigger input (<exp trig>).

- 0x0003 = [extern exposure control]: The exposure time is defined by the pulse length at the trigger input (<exp trig>). The delay and exposure time values defined by the set/request delay and exposure command are ineffective. (Exposure time length control is also possible for double image mode; exposure time of the second image is given by the readout time of the first image.)

Note: Modes [extern exposure & software trigger] and [extern exposure control], depend also on the selected acquire mode. A trigger edge at the trigger input (<exp trig>) will be effective or not (see also SetAcquireMode.vi). A software trigger however will always be effective independent of the state of the <acq enbl> input.



hout Handle output

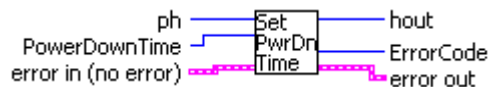


error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

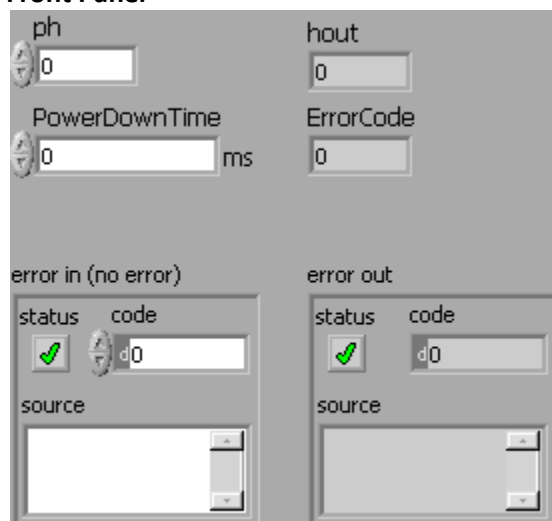
4.3.16. SetUserPowerDownTime.vi

Sets the power down time in milliseconds, for the user power down mode. Use SetPowerDownMode to enable the user power down mode.







Connector Pane



Front Panel



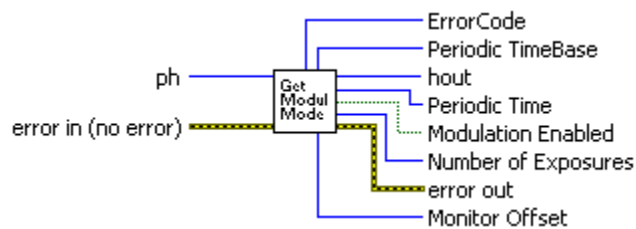
Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
-  **PowerDownTime** The time to power down, in milliseconds, for user power down mode
-  **hout** Handle output
-  **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
-  **ErrorCode**

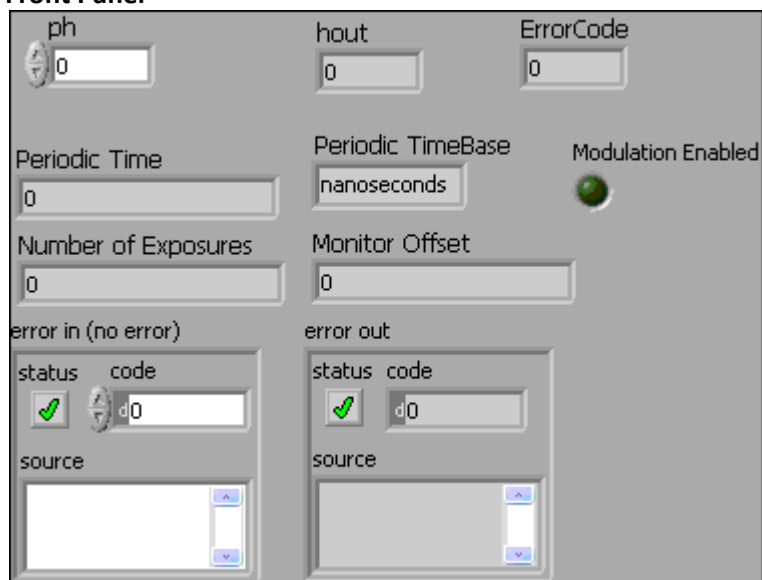
4.3.17. GetModulationMode.vi

Gets the modulation parameters for cameras with modulation capability. Use "IMAQGetDescriptionEx.vi" to determine if the camera has modulation capability.

Connector Pane



Front Panel



Controls and Indicators

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

ph Handle for the camera

error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

Periodic Time Period (cycle time) of the modulation, in timebase units.



Number of Exposures Number of exposures per frame.



Periodic TimeBase Time base (units of time) for the period setting.

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds



Monitor Offset Time between start of exposures and the monitor output active, in nanoseconds. Can be negative to start the monitor signal before the exposure.



Modulation Enabled Indicates if the modulation function is currently on or off.



hout Handle output

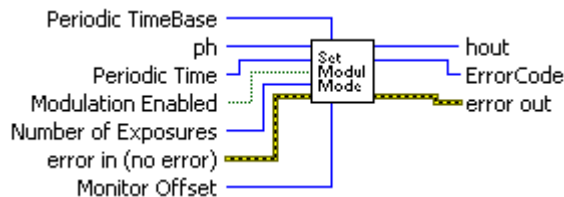


ErrorCode

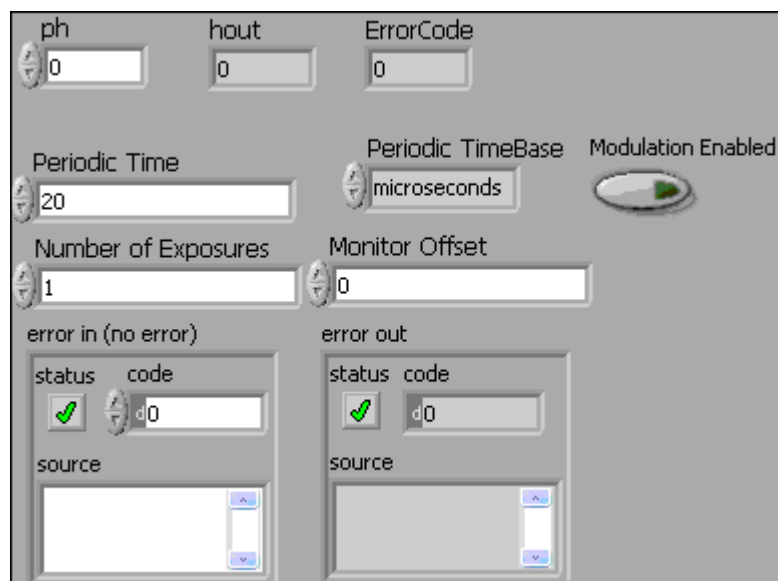
4.3.18. SetModulationMode.vi

Sets the modulation parameters for cameras with modulation capability. Use "IMAQGetDescriptionEx.vi" to determine if the camera has modulation capability.

Connector Pane



Front Panel



Controls and Indicators



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Periodic Time Period (cycle time) of the modulation, in timebase units.



Number of Exposures Number of exposures per frame.



Periodic TimeBase Time base (units of time) for the modulation period.

0 - Nanoseconds

1 - Microseconds

2 - Milliseconds

I32 **Monitor Offset** Time between start of exposures and the monitor output active, in nanoseconds. Can be negative to start the monitor signal before the exposure.

TF **Modulation Enabled** Turns the modulation function on or off.

U64 **ph** Handle for the camera

8 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

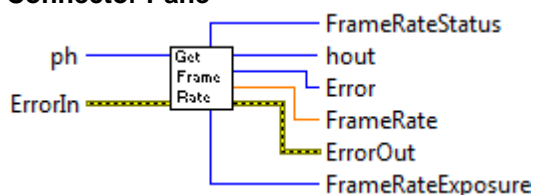
U64 **hout** Handle output

I32 **ErrorCode**

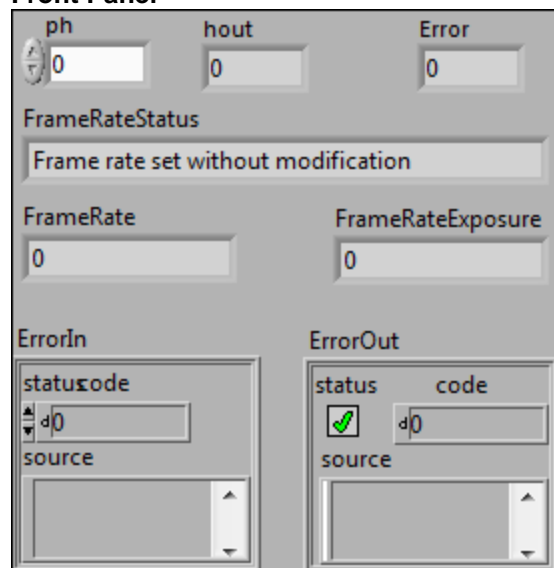
4.3.19. GetFrameRate.vi

Finds the current frame rate. Indicates if the specified frame rate has been modified to match system parameters. Returns the actual frame rate and exposure time.

Connector Pane



Front Panel



Controls and Indicators



ph Handle in.



FrameRateStatus Status of the last SetFrameRate command.



FrameRate Current frame rate, in Hz.



FrameRateExposure Current exposure time for frame rate mode, in ns.



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.



ErrorOut The error in cluster can accept error information wired from VIs previously called.



Error Error code from driver

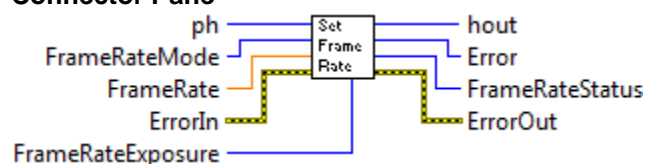


hout Handle output

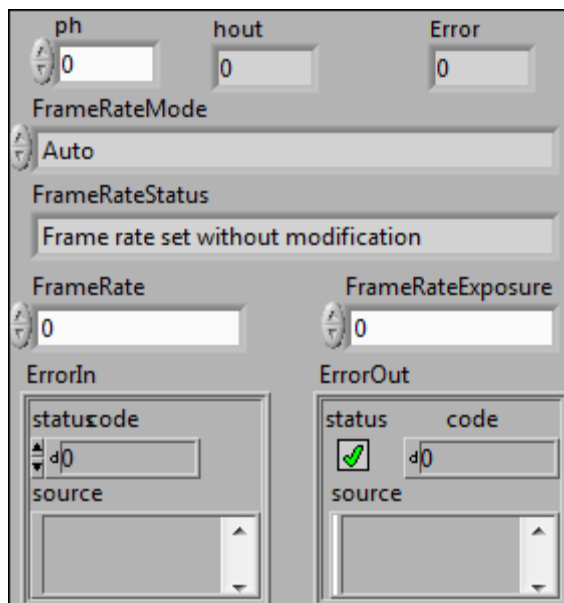
4.3.20. SetFrameRate.vi

Sets the frame rate (pco.dimax ONLY!). Indicates if the specified frame rate has been modified to match system parameters. Returns the actual frame rate and exposure time.

Connector Pane



Front Panel



Controls and Indicators



ph Handle input



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.



FrameRateStatus Status of the last SetFrameRate command.



FrameRate Desired frame rate, in Hz.



FrameRateExposure Desired exposure time for frame rate mode, in ns.



FrameRateMode Determines how the specified frame rate is adapted to the system capabilities. Frame rate and exposure time may be modified based on sensor parameters and clock rates.

Auto: Camera decides which of frame rate or exposure is modified

Frame rate priority: Exposure will be modified if necessary

Exposure time priority: Frame rate will be modified if necessary

Strict: Frame rate and exposure will not be modified, but an error will result if parameters are not possible.



Error



hout

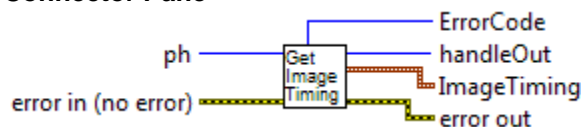


ErrorOut The **error in** cluster can accept error information wired from VIs previously called.

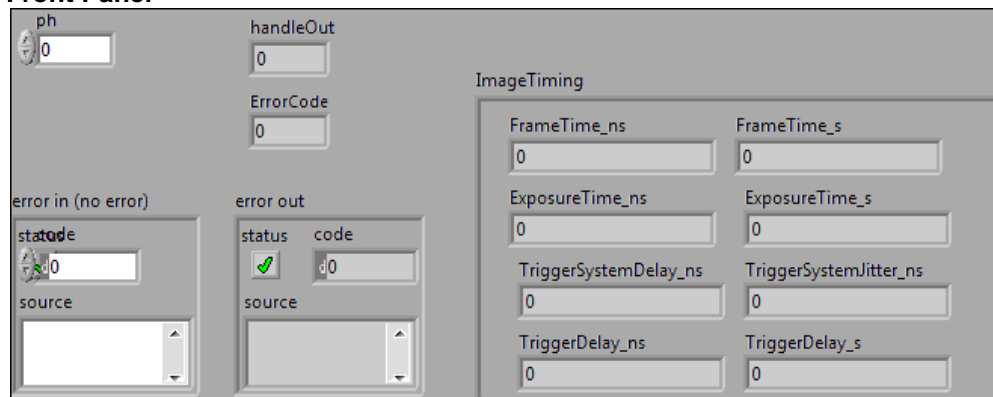
4.3.21. GetImageTiming.vi

Returns detailed timing information, including system delays and jitter. Can be used to make more accurate determination of the exposure timing.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



handleOut Handle to the camera



ImageTiming Detailed information about the exposure and delay timing, including internal delays and jitter.








FrameTime_ns Time required, with FrameTime_s, to acquire one frame



FrameTime_s Time required, with FrameTime_ns, to acquire one frame



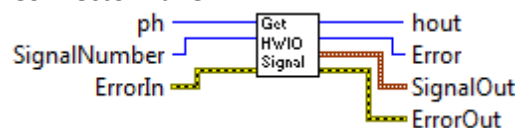
ExposureTime_ns This + ExposureTime_s = exposure time

-  **ExposureTime_s** This + ExposureTime_ns = exposure time
-  **TriggerSystemDelay_ns** Minimum internal trigger system delay in ns
-  **TriggerSystemJitter_ns** Maximum possible jitter, +/- in ns
-  **TriggerDelay_ns** This + TriggerDelay_s = total delay, including programmed and system delays
-  **TriggerDelay_s** This + TriggerDelay_ns = total delay, including programmed and system delays

4.3.22. GetHWIOSignal.vi

Provides details of the selected hardware signal, including signal names and I/O modes

Connector Pane



Front Panel

The LabVIEW interface is divided into several sections:

- ph**: A numeric control with a value of 0.
- hout**: A numeric control with a value of 0.
- SignalNumber**: A numeric control with a value of 0.
- Error**: A numeric control with a value of 0.
- SignalOut**: A structure control with the following fields:
 - Size**: 60
 - SignalNumber**: 0
 - Enabled**: 0
 - Type**: 0
 - Polarity**: 0
 - FilterSettings**: 0
 - Selected**: 0
 - Reserved**: 0
 - Reserved 2**: 0
 - Reserved 3**: 0
 - Reserved 4**: 0
 - Reserved 5**: 0
 - Reserved 6**: 0
 - Reserved 7**: 0
 - Reserved 8**: 0
 - Reserved 9**: 0
 - Reserved 10**: 0
 - Reserved 11**: 0
 - Reserved 12**: 0
- ErrorIn**: A status indicator with a green checkmark, code 0, and source field.
- ErrorOut**: A status indicator with a green checkmark, code 0, and source field.

Controls and Indicators

- U64** **ph** Handle to an open camera connection
- U16** **SignalNumber** Number of the HWIO signal to inquire about
- U16** **SignalOut** Description of the HWIO signal
- U16** **Size** Size of structure in bytes (do not modify)
 - U16** **SignalNumber** Number of the HWIO signal
 - U16** **Enabled** State of the HWIO signal:
 - 0x00: Signal is disabled
 - 0x01: Signal is enabled
 - U16** **Type** Flags showing which signal type is selected:
 - 0x01: TTL
 - 0x02: High Level TTL
 - 0x04: Contact Mode
 - 0x08: RS485 differential



Polarity Flags showing which signal polarity is selected:

- 0x01: High level active
- 0x02: Low Level active
- 0x04: Rising edge active
- 0x08: Falling edge active



FilterSettings Flags showing the filter option which is selected:

- 0x01: Filter can be switched off ($t > \sim 65\text{ns}$)
- 0x02: Filter can be switched to medium ($t > \sim 1\mu\text{s}$)
- 0x04: Filter can be switched to high ($t > \sim 100\text{ms}$)



Selected If HWIOSignaldescription shows more than one SignalNames, this parameter indicates which signal is active, e.g. 'Status Busy' or 'Status Exposure'.



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.



Error Error number from driver



hout Handle to an open camera connection (same as input handle)

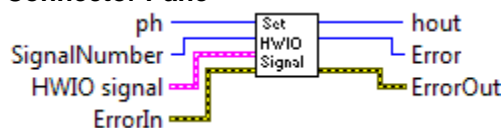


ErrorOut The **error in** cluster can accept error information wired from VIs previously called.

4.3.23. SetHWIOSignal.vi

Controls the properties of the selected hardware signal, including modes and polarities

Connector Pane



Front Panel

Controls and Indicators



ph Handle to a open camera connection



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.









SignalNumber




HWIO signal Cluster of parameters describing the current configuration of the selected


HWIO signal

-  **SignalNumber** Number of the HWIO signal selected
-  **Enabled** True if signal enabled, false if signal disabled
-  **Type** Signal level
-  **Polarity** Sense of signal voltage, or edge if edge sensitive
-  **FilterSettings** Level of filtering on signal
-  **Selected** Selects which variant of the signal is used, if signal has more than one variant


 **Error**

 **hout** Handle to a open camera connection (same as input handle)

 **ErrorOut** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

 **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **code** The **code** input identifies the error or warning.

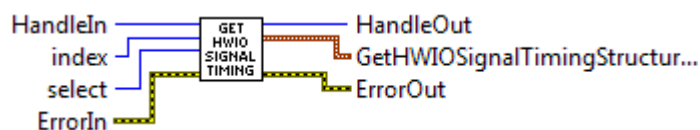
The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

4.3.24. GetHWIOSignalTiming.vi

Queries the functionality on any hardware I/O signal that has multiple modes of operation.



The screenshot displays the LabVIEW front panel for the pco.camera interface. It includes several control fields (indicated by a small 'A' icon) and indicator fields (indicated by a small 'I' icon).

- HandleIn**: A numeric control field with the value 0.
- index**: A numeric control field with the value 0003.
- select**: A numeric control field with the value 0000.
- HandleOut**: A numeric indicator field with the value 0.
- Error**: An error indicator field showing a green checkmark and the value 0.
- GetHWIOSignalTimingStructure Out**: A cluster of fields:
 - code**: Numeric control field (0).
 - parameter**: Numeric control field (0).
 - checksum**: Numeric control field (0).
 - length**: Numeric control field (33).
 - reserved 0**: Numeric control field (0).
 - index**: Numeric control field (0).
 - reserved 1**: Numeric control field (0).
 - select**: Numeric control field (0).
 - reserved 2**: Numeric control field (0).
 - type**: Numeric control field (0).
 - reserved 3**: Numeric control field (0).
- ErrorIn**: An error cluster containing:
 - status**: Indicator with a green checkmark.
 - code**: Numeric indicator field (40).
 - source**: A list box showing 'Layer: DLL, Device: SDK'.
- ErrorOut**: An error cluster containing:
 - status**: Indicator with a green checkmark.
 - code**: Numeric indicator field (40).
 - source**: A list box showing 'Layer: DLL, Device: SDK'.



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.



HandleIn Handle to the selected camera



index Selects the HWIO signal to be queried



select On HWIO signals that have multiple functions, this selects the function of the signal. e.g. exposure out or busy out.



ErrorOut The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



GetHWIOSignalTimingStructure Out



code Low level control code



length Length of message



index Index of signal queried



select Selected function of this signal

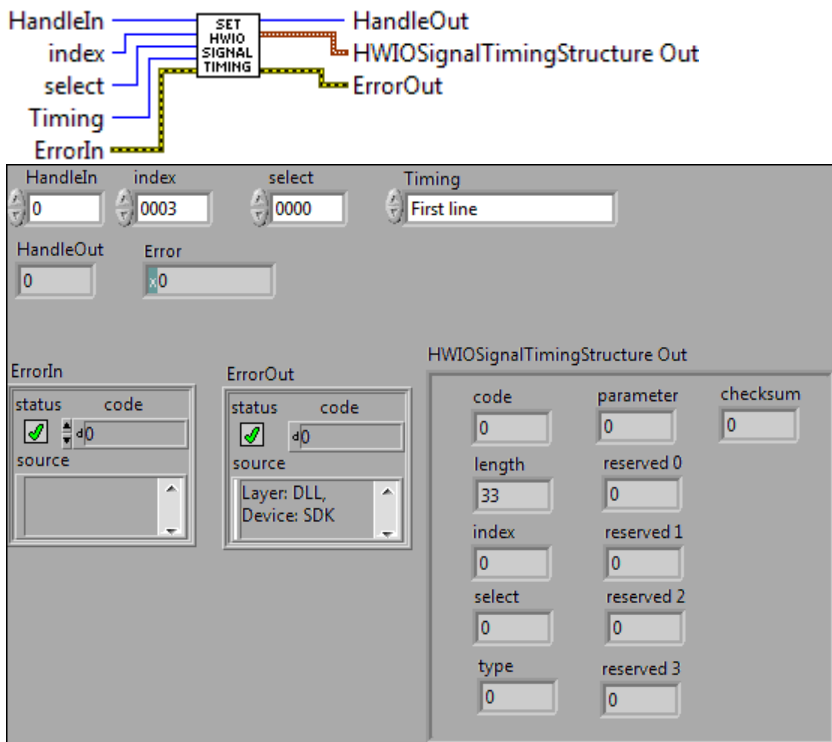


type Type of signal

- parameter** Timing details
 - reserved 0**
 - reserved 1**
 - reserved 2**
 - reserved 3**
 - checksum**
- HandleOut** Handle to selected camera

4.3.25. SetHWIOSignalTiming.vi

Selects the functionality on any hardware I/O signal that has multiple modes of operation This can be used to configure output signals on the pco.edge.



- ErrorIn** The **error in** cluster can accept error information wired from VIs previously called.
- HandleIn** Handle to selected camera



Timing Selects the timing options for the HWIO signal



index Selects the HWIO signal to be configured



select On HWIO signals that have multiple functions, this selects the function of the signal. e.g. exposure out or busy out.



ErrorOut The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



HWIOSignalTimingStructureOut



code Low level control command



length Length of command message



index Selected signal



select Selected function



type Type of signal



parameter Timing options



reserved 0



reserved 1



reserved 2



reserved 3



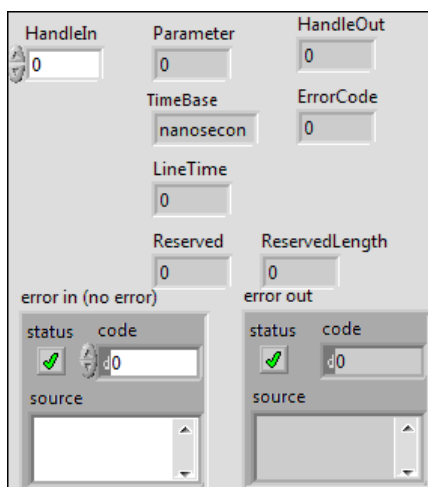
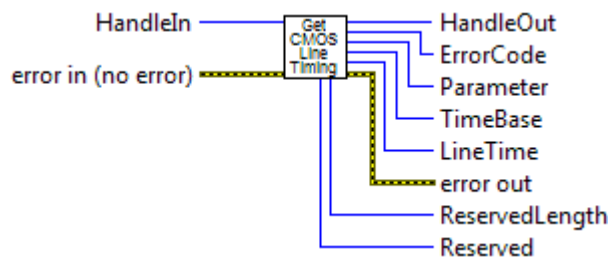
checksum



HandleOut Handle to selected camera

4.3.26. GetCMOSLineTiming.vi

Sets the timing for an individual line in sCMOS cameras. This enables the camera scan rate to be matched to a moving source.



U64 **HandleIn** Handle to the selected camera

error in (no error) The **error in** cluster can accept error information wired from VIs previously called.

I32 **ErrorCode**

U64 **HandleOut**

error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

U16 **Parameter** Displays the operation of the lime timing mode

CMOS_LINETIMING_PARAM_OFF = 0
CMOS_LINETIMING_PARAM_ON = 1

U16 **TimeBase** Time base (units of time) for the exposure setting.

0 - Nanoseconds
1 - Microseconds
2 - Milliseconds

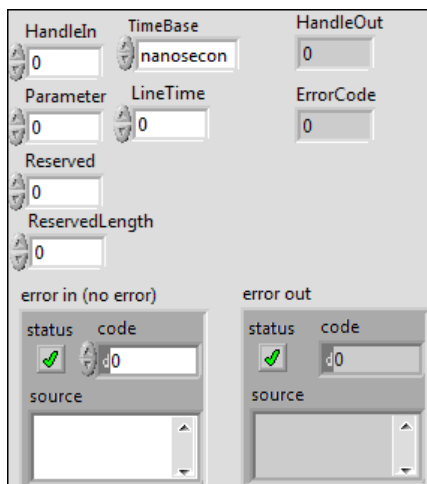
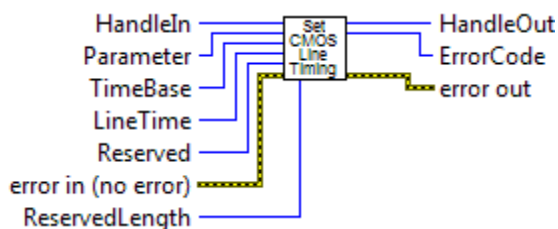
U32 **LineTime** Time of one line, in time base units.

U32 **Reserved** Reserved for future use

U16 **ReservedLength** Reserved for future use

4.3.27. SetCMOSLineTiming.vi

Sets the timing for an individual line in sCMOS cameras. This enables the camera scan rate to be matched to a moving source.



U64 **HandleIn** Handle to the selected camera

U16 **Parameter** Controls the operation of the lime timing mode

CMOS_LINETIMING_PARAM_OFF = 0
CMOS_LINETIMING_PARAM_ON = 1

U16 **TimeBase** Time base (units of time) for the exposure setting.

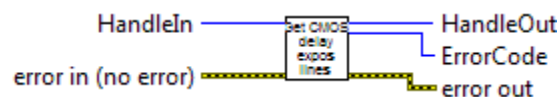
0 - Nanoseconds
1 - Microseconds
2 - Milliseconds

U32 **LineTime** Time of one line, in time base units.

**Reserved** Reserved for future use**ReservedLength** Length of reserved parameter list**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called**ErrorCode****HandleOut****error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

4.3.28. GetCMOSLineExposureDelay.vi





This command returns the exposure and delay time for a frame. It is only valid when the line timing parameter is set to CMOS_LINETIMING_PARAM_ON.



The front panel displays the following controls:

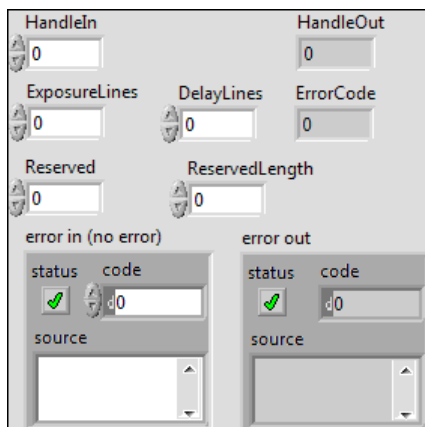
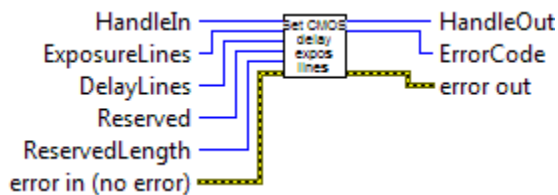
- HandleIn:** A numeric control set to 0.
- ExposureLines:** A numeric control set to 0.
- Reserved:** A numeric control set to 0.
- DelayLines:** A numeric control set to 0.
- ReservedLength:** A numeric control set to 0.
- errorCode:** A numeric control set to 0.
- error in (no error):** A cluster containing a status indicator (green checkmark), a code numeric control (set to 0), and a source list box.
- error out:** A cluster containing a status indicator (green checkmark), a code numeric control (set to 0), and a source list box.








**HandleIn** Handle for the selected camera**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called**ErrorCode** 0 if no error, otherwise an error code appears here**HandleOut** Handle of the selected camera**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

-  **ExposureLines** Number of lines required for exposure
-  **DelayLines** Number of lines between consecutive frames
-  **Reserved** Reserved for future features
-  **ReservedLength** Length in bytes of the reserved structure

4.3.29. SetCMOSLineExposureDelay.vi

This command sets the exposure and delay time for a frame. It is only available when the line timing parameter is set to CMOS_LINETIMING_PARAM_ON.



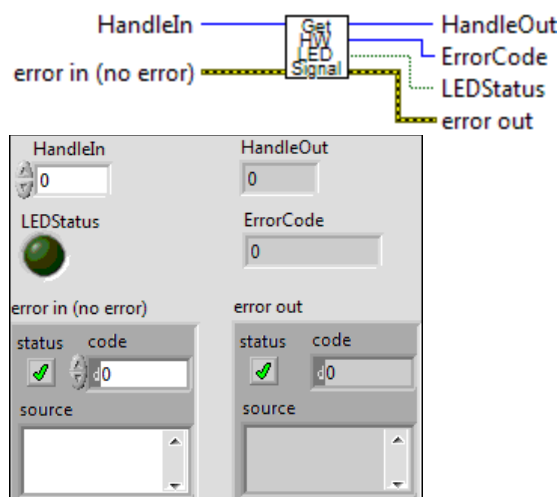
-  **HandleIn** Handle for the selected camera
-  **ExposureLines** Number of lines required for exposure
-  **DelayLines** Number of lines between consecutive frames
-  **Reserved** Reserved for future features
-  **ReservedLength** Length in bytes of the reserved structure
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called.
-  **ErrorCode** 0 if no error, otherwise an error code appears here

U64 **HandleOut** Handle of the selected camera

E64 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

4.3.30. GetHWLEDSignal.vi

Gets the status of the LED at the back of the pco.edge camera



U64 **HandleIn** Handle to the requested camera

E64 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called

U64 **HandleOut** Handle to the requested camera

E64 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

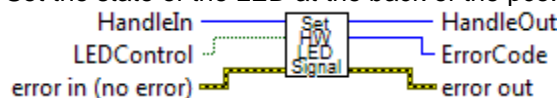
I32 **ErrorCode** 0 if no error, otherwise an error code will appear here

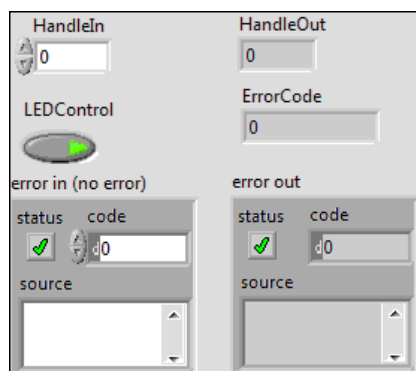
TF **LEDStatus** LED enabled state:

ON = Enabled
OFF = Disabled

4.3.31. SetHWLEDSignal.vi

Set the state of the LED at the back of the pco.edge camera





HandleIn Handle to the requested camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called.



LEDControl Enable or disable LED operation:

ON = Enabled

OFF = Disabled



HandleOut Handle to the requested camera



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode 0 if no error, otherwise an error code will appear here

4.4. Storage.llb

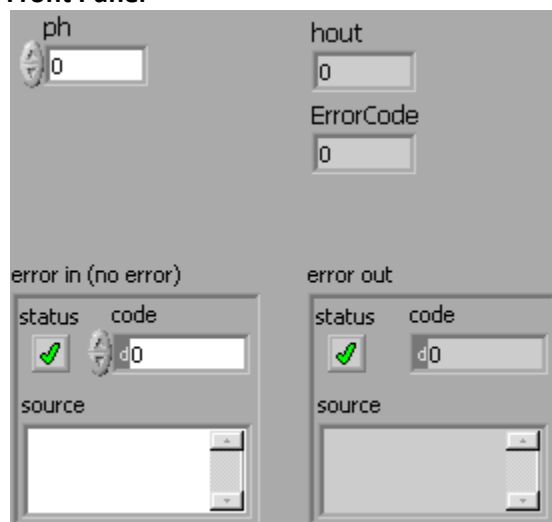
4.4.1. ClearRAMSegment.vi

Clears the active RAM segment. All previously recorded images are lost.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



Error out

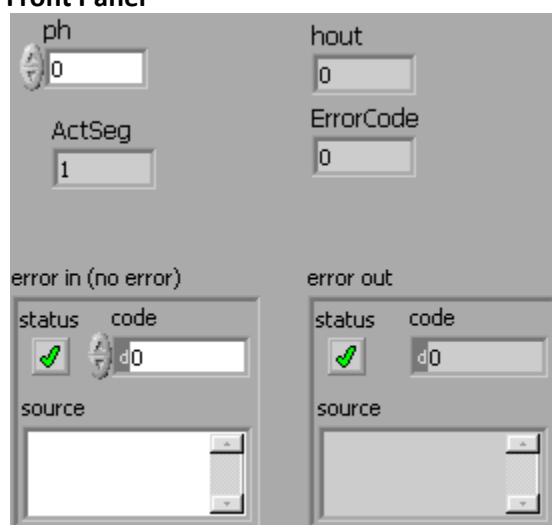
4.4.2. GetActiveRAMSegment.vi

Determines which camRAM segment is currently active.







Connector Pane



Front Panel



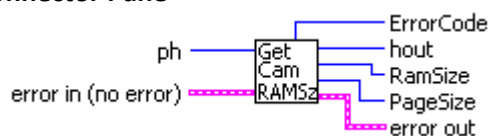
Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
-  **hout** Handle output
-  **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
-  **ErrorCode**
-  **ActSeg** Number of the active segment

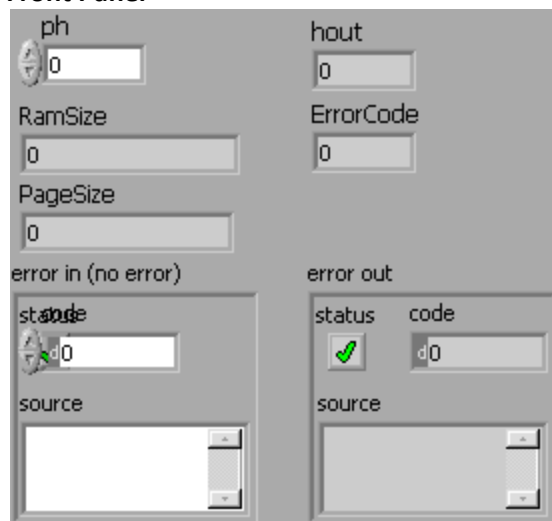
4.4.3. GetCameraRAMSize.vi

Finds the size of the camera RAM (camRAM) in pages. One page is the smallest unit for RAM segmentation as well as for storing images. Segment sizes can only be configured as multiples of pages. The size reserved for one image is also calculated as multiples of whole pages, therefore there may be some unused RAM memory if the page size is not exactly a multiple of the image size. The number of pages needed for one image depends on the image size (Xres x Yres) divided by the pixels per page (page size). Every begun page size has to be considered, so if you have 50.6 pages for an image you will need 51 pages for this image. With this value of 'pages per image' you can calculate the number of images fitting into the segment.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



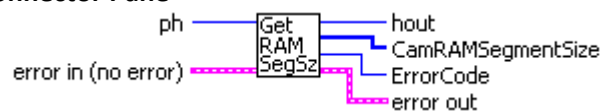
error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**hout** Handle output**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.**ErrorCode****RamSize** Total size of camera RAM, in pages.**PageSize** Size of one page, in pixels. This number will determine how many pages will be needed to store one image of X x Y pixels.

4.4.4. GetCamRAMSegmentSize.vi

Finds the RAM segment sizes in pages. A size of zero pages indicates that the segment will not be used.

Connector Pane



Front Panel



Controls and Indicators

**ph** Handle for the camera**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode

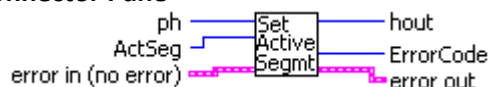


CamRAMSegmentSize Array of RAM segment sizes in pages, one element per segment. Element 0 is the size of segment 1, element 1 is the size of segment 2, etc.

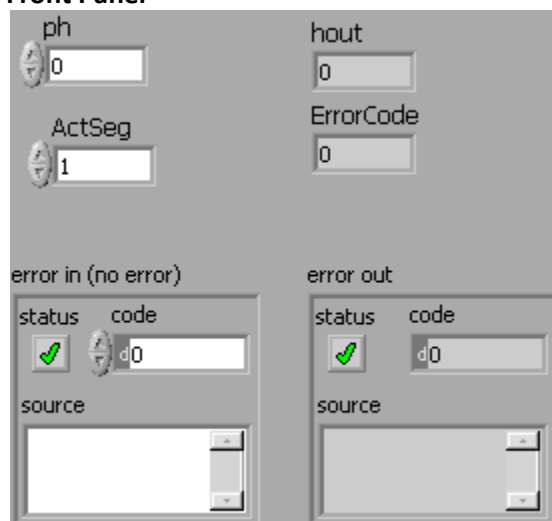
4.4.5.SetActiveRAMSegment.vi

Selects one of 4 camRAM segments to be the active segment

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

U16 **ActSeg** Selects the number of the active segment. Valid numbers are integers from 1 to 4.

U64 **hout** Handle output

E11 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

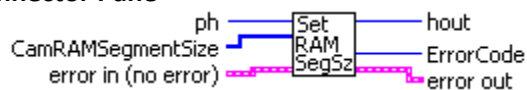
I32 **ErrorCode**

4.4.6. SetCamRAMSegmentSize.vi

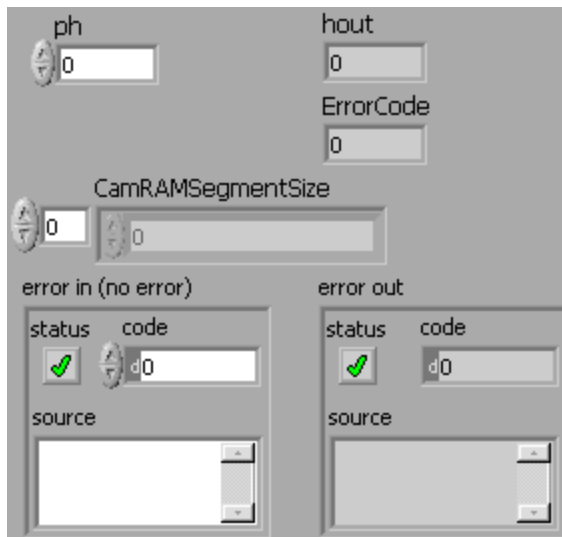
Sets the sizes of RAM segments

- The sum of all segment sizes must not be larger than the total size of the RAM (as multiples of pages)
- A single segment size can have the value 0x0000, but the sum of all 4 segments must be greater than 0x0000.
- The command will be rejected, if Recording State is [run]
- The function will result in all segments to be cleared. All images recorded before are lost!

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



CamRAMSegmentSize Array of RAM segment sizes in pages, one element per segment. Element 0 is the size of segment 1, element 1 is the size of segment 2, etc.

- The sum of all segment sizes must not be larger than the total size of the RAM (as multiples of pages)
- A single segment size can have the value 0x0000, but the sum of all 4 segments must be greater than 0x0000.
- The command will be rejected, if Recording State is [run]
- The function will result in all segments to be cleared. All images recorded before are lost!



hout Handle output



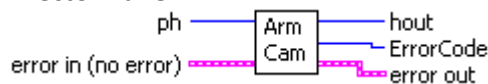
error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

4.5. RecordingControl.llb

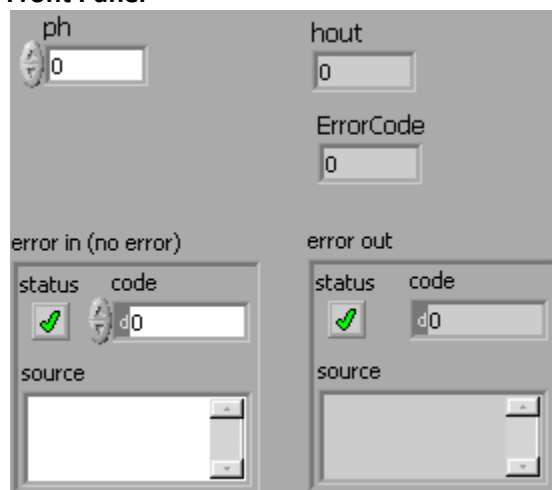
4.5.1. ArmCamera.vi

Arms the camera. Arming loads the desired settings in preparation for the start of a new recording. Settings do not take effect until after an "ArmCamera" command.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

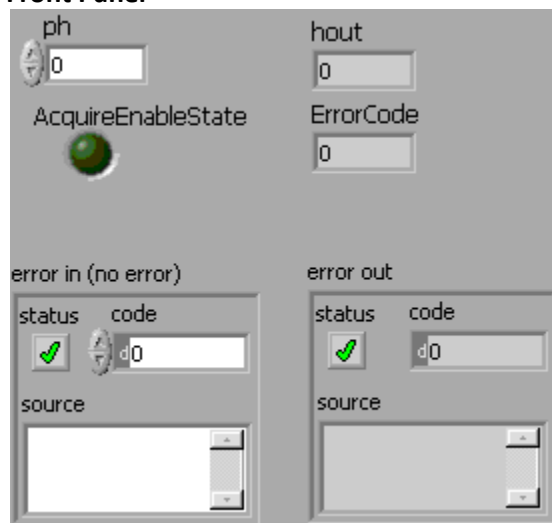
4.5.2. GetAcquireEnableSignalStatus.vi

Finds the status of the "acquire enable" signal. If this signal is TRUE and the camera is in "external acquisition control" mode, images acquisition is enabled.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



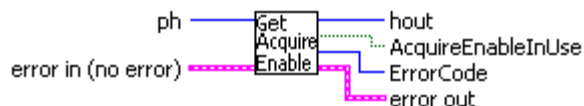
AcquireEnableState Finds the status of the "acquire enable" signal. If this signal is TRUE and the camera is in "external acquisition control" mode, images acquisition is enabled. The state of the "acquire enable" signal depends on the input voltage level and the dip switch settings on the pco.power unit.

- input signal = HIGH, DIP switch = HIGH: Status = TRUE
- input signal = HIGH, DIP switch = LOW: Status = FALSE
- input signal = LOW, DIP switch = HIGH: Status = FALSE
- input signal = LOW, DIP switch = LOW: Status = TRUE

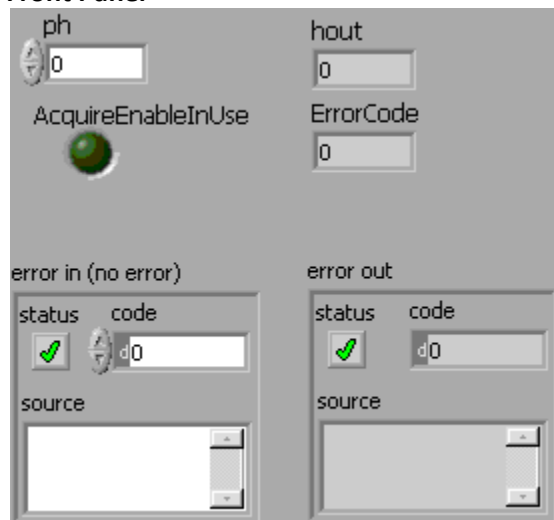
4.5.3. GetAcquireMode.vi

Determines if the "acquire enable" control is currently in use.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs

previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



AcquireEnableInUse Indicates if the camera is using the "acquire enable" signal.

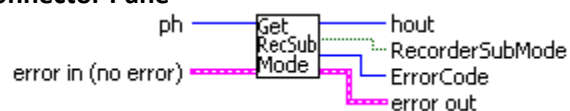
FALSE - The acquire enable signal is not in use

TRUE - The camera is using the acquire enable signal to control acquisition.

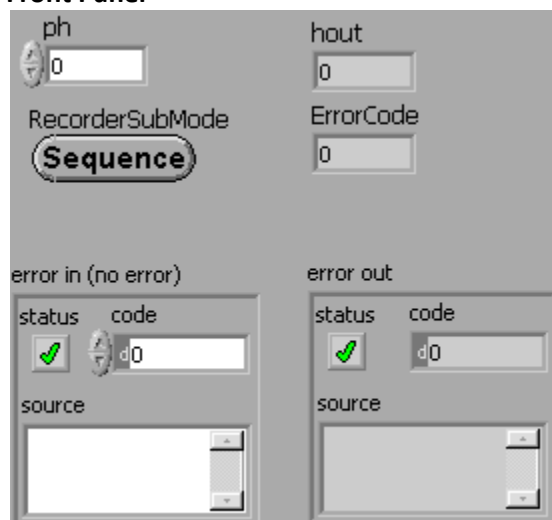
4.5.4. GetRecorderSubmode.vi

Finds the current recorder submode. Valid modes are "Sequence" and "Ring Buffer".

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



RecorderSubMode Current recorder submode.

FALSE - The submode is Sequence

TRUE - The submode is Ring Buffer

4.5.5. GetRecordingState.vi

Finds the recording state of the camera.

The recording state controls the status of the camera. If the recording status is [run], images can be started by exposure trigger and <acq enbl>. If the recording status is cleared or stopped, all image readout or exposure sequences are stopped and the sensors (CCD or CMOS) are running in a special idle mode to prevent dark charge accumulation. The recording status has the highest priority compared to functions like <acq enbl> or exposure trigger.

The recording status is controlled by software command: set recording status = [run]

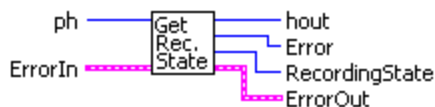
The recording status is cleared by:

Powering ON the camera

Software command: set recording status = [stop]

Software command: reset all settings to default values

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



RecordingState Current recording state

0 - Recording stopped

1 - Camera is recording



Error



hout Camera handle output



ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

4.5.6. GetStorageMode.vi

Finds the current recording mode of the camera. Possible values are Recorder mode and FIFO mode

Recorder Mode:

Images are recorded and stored within the internal camera memory (camRAM)

Live View transfers the most recent image to the PC (for viewing / monitoring)

Indexed or total readout of images after the recording has been stopped

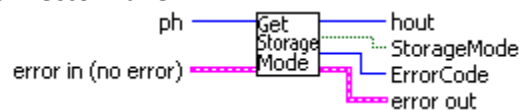
FIFO Buffer mode:

All images taken are transferred to the PC in chronological order

Camera memory (camRAM) is used as huge FIFO buffer to bypass short bottlenecks in data transmission

If buffer overflows the oldest images are overwritten

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



StorageMode Current storage mode.

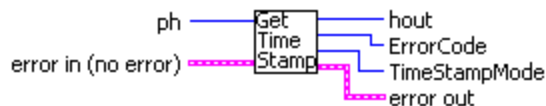
FALSE - Camera is in Recorder mode

TRUE - Camera is in FIFO mode

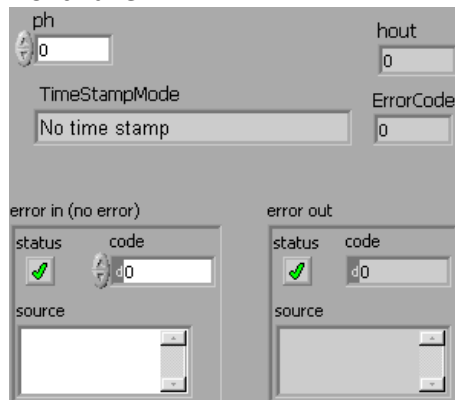
4.5.7. GetTimeStampMode.vi

Indicates the state of the time stamping mode. Time stamping may be disabled, or encoded as BCD or BCD and ASCII text.

Connector Pane



Front Panel



Controls and Indicators

- U64

ph Handle for the camera
- E8

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- U64

hout Handle output
- E8

error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
- I32

ErrorCode
- U16

TimeStampMode Current time stamp mode. Possible values are:

0 - Time stamp mode disabled.

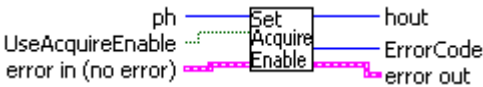
1 - Binary Coded Decimal (BCD) time stamp in the first 14 pixels

2 - BCD time stamp in the first 14 pixels + ASCII text

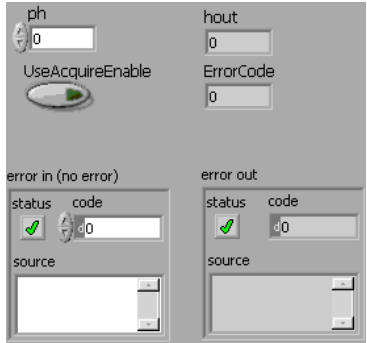
4.5.8.SetAcquireMode.vi

Enables or disables the use of the "acquire enable" signal to control acquisition.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



UseAcquireEnable Enables or disables the use of the "acquire enable" signal to control acquisition.

FALSE - Disables the use of the "acquire enable" signal

TRUE - Enables the use of the "acquire enable" signal



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

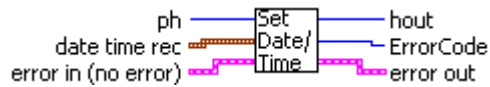


ErrorCode

4.5.9.SetDateTime.vi

Sets the camera's internal clock. Once set, the clock can be used for precision time stamping. The clock runs continuously while the camera is powered up, but the time is not maintained when the power is off.




Connector Pane



Front Panel



Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
-  **date time rec** LabVIEW date/time cluster, compatible with the LabVIEW "Seconds to

Date/Time.vi"



second



minute



hour



day of month



month



year



day of week



day of year



is DST



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

4.5.10. SetRecorderSubmode.vi

Sets the recorder submode. Valid modes are "Sequence" and "Ring Buffer". These submodes only apply to the "Recorder" storage mode. They make no difference to the FIFO mode.

Sequence:

Recording is stopped when the allocated buffer is full

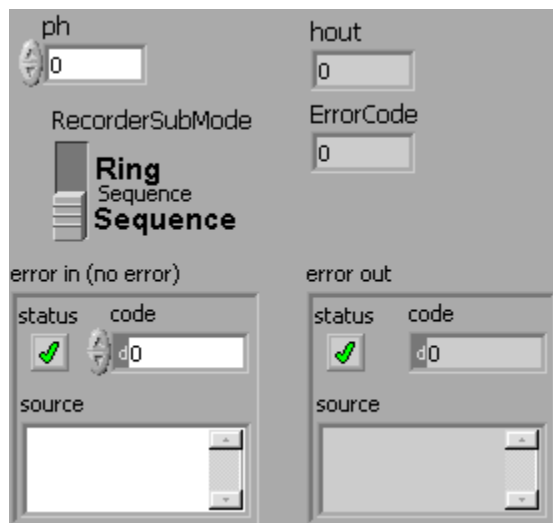
Ring Buffer:

Camera records continuously into ring buffer. If the allocated buffer overflows, the oldest images are overwritten. Recording is stopped by software or disabling acquire signal (<acq enbl>)

Connector Pane



Front Panel



Controls and Indicators

U64 **ph** Handle for the camera

ST **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

TF **RecorderSubMode** Required recorder submode.

FALSE - Sets the sequence mode

TRUE - Sets the Ring Buffer

U64 **hout** Handle output

ST **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

I32 **ErrorCode**

4.5.11. SetRecordingState.vi

Sets the recording state of the camera.

The recording state controls the status of the camera. If the recording status is [run], images can be started by exposure trigger and <acq enbl>. If the recording status is cleared or stopped, all image readout or exposure sequences are stopped and the sensors (CCD or CMOS) are running in a special idle mode to prevent dark charge accumulation. The recording status has the highest priority compared to functions like <acq enbl> or exposure trigger.

The recording status is controlled by software command: set recording status = [run]

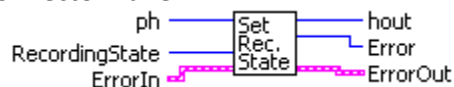
The recording status is cleared by:

Powering ON the camera

Software command: set recording status = [stop]

Software command: reset all settings to default values

Connector Pane



Front Panel



Controls and Indicators



RecordingState Set the required recording state

0 - Stop recording

1 - Start recording



ph Handle to a previously opened camera



ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Error



hout Handle to camera



ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

4.5.12. SetStorageMode.vi

Sets the recording mode of the camera. Possible values are Recorder mode and FIFO mode

Recorder Mode:

Images are recorded and stored within the internal camera memory (camRAM)

Live View transfers the most recent image to the PC (for viewing / monitoring)

Indexed or total readout of images after the recording has been stopped

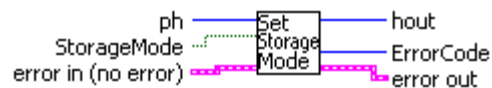
FIFO Buffer mode:

All images taken are transferred to the PC in chronological order

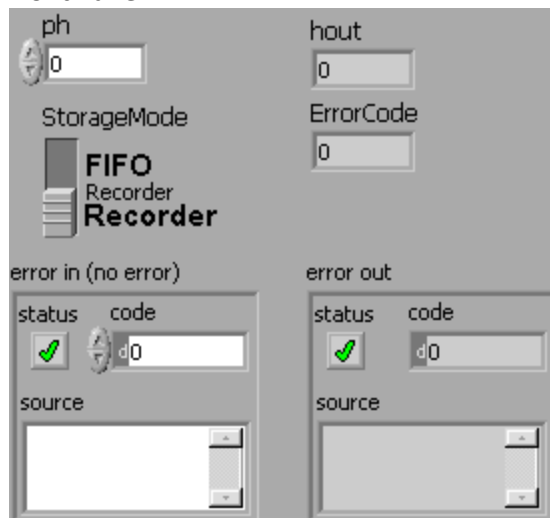
Camera memory (camRAM) is used as huge FIFO buffer to bypass short bottlenecks in data transmission

If buffer overflows the oldest images are overwritten

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



StorageMode Sets the storage mode of the camera.

FALSE - Selects Recorder mode

TRUE - Selects FIFO mode



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode

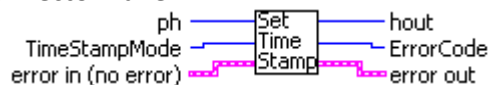
4.5.13. SetTimeStampMode.vi

Set mode of the timestamp function. Time stamping writes the continuous image number and date / time information with a resolution of 10 μ s direct into the raw image data. The first 14 pixels (top left corner) are used to hold this information. The numbers are coded in BCD with one byte per pixel, which means that every pixel can hold 2

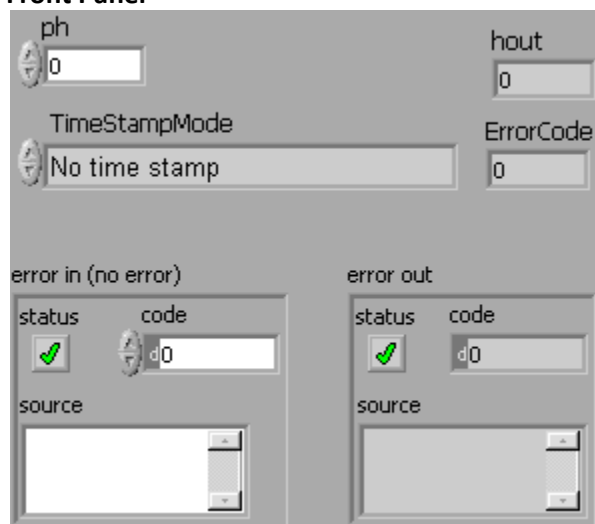
digits. If the pixels have more resolution than 8 bits, then the BCD digits are left aligned (MS bit) and the lower bits padded with zeroes. In addition to the 14 pixel binary stamp, the information can be written in ASCII text for direct inspection. An 8 by 8 pixel array is used to hold the ASCII data. The digits are displayed below the BCD coded line.

Format of BCD coded pixels:

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



TimeStampMode Sets the time stamp mode. Possible values are:

0 - Time stamp mode disabled.

1 - Binary Coded Decimal (BCD) time stamp in the first 14 pixels

2 - BCD time stamp in the first 14 pixels + ASCII text



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



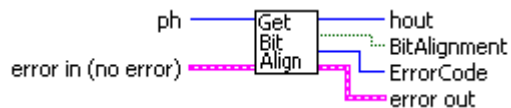
ErrorCode

4.6. BufferData.llb

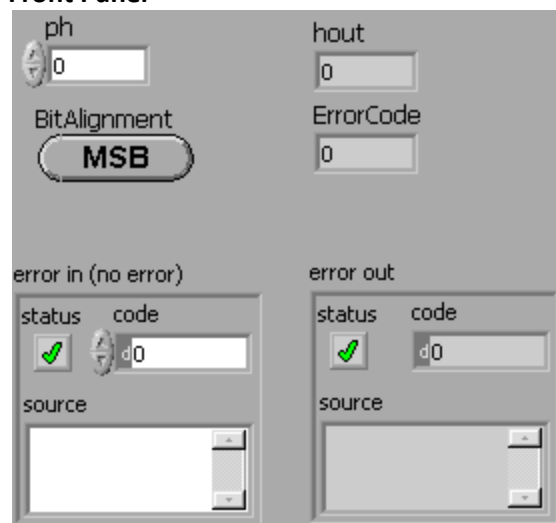
4.6.1. GetBitAlignment.vi

Determines the bit alignment that is currently in use for storing images. For image data less than 16 bits, the data will be aligned to either the MS bit or the LS bit, and the remaining bits padded with zeroes.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



BitAlignment Describes the current setting for the bit alignment.

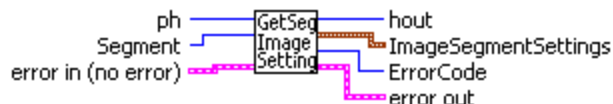
FALSE - Values are aligned to the MS bit and the LS bits are zero

TRUE - Values are aligned to the LS bit and the MS bits are zero

4.6.2. GetImageSegmentSettings.vi

Finds information about the images in the specified segment. The horizontal and vertical resolution and binning as well as the region of interest settings are returned.

Connector Pane



Front Panel

Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Segment Number of the segment to get settings from. Values of 1 to 4 are allowed.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



ImageSegmentSettings Settings for the images in the selected segment



XRes Horizontal Resolution of the images in the segment



YRes Vertical Resolution of the images in the segment



BinHorz Horizontal binning of the images in the segment



BinVert Vertical binning of the images in the segment



ROI X0 Leftmost horizontal ROI setting



ROI Y0 Upper vertical ROI setting



ROI X1 Rightmost horizontal ROI setting



ROI Y1 Lower vertical ROI setting

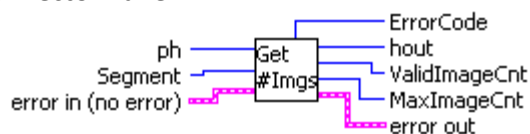
4.6.3. GetNumberOfImagesInSegment.vi

Get the number of valid images within the segment. The operation is slightly different due to the selected storage mode:

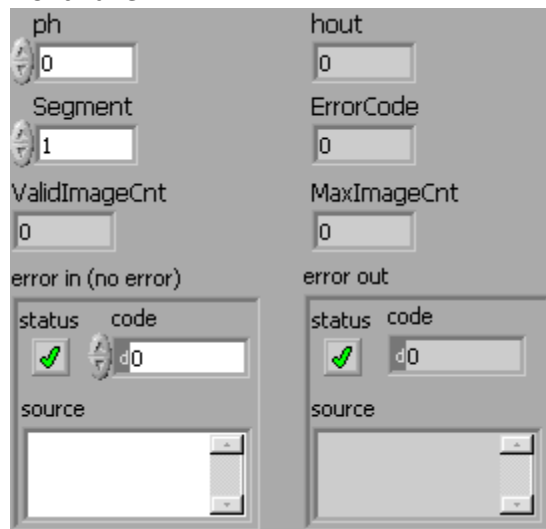
In [recorder mode], if recording is not stopped and in [FIFO buffer mode] the number of images is dynamic due to read and write accesses to the camera RAM.

In [recorder mode] and recording is stopped, the number is fixed.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Segment Segment to find the image count. Valid numbers are 1 to 4.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



ValidImageCnt Number of valid images in this segment

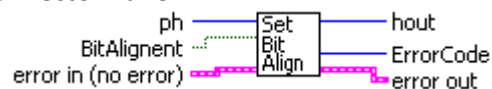


MaxImageCnt Maximum number of valid images that can be stored in this segment

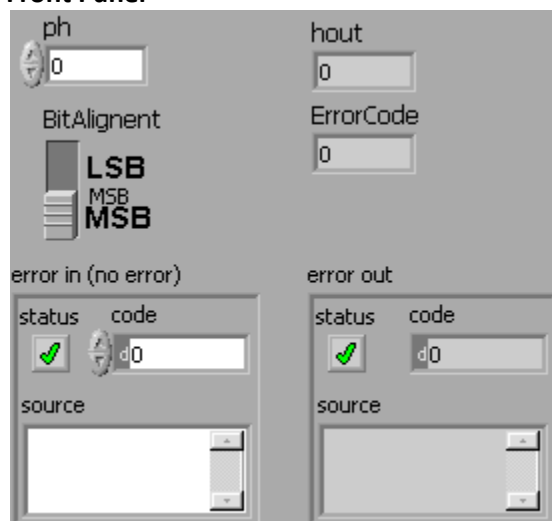
4.6.4.SetBitAlignment.vi

Sets the bit alignment that is used for storing images. For image data less than 16 bits, the data will be aligned to either the MS bit or the LS bit, and the remaining bits padded with zeroes.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



BitAlignent Sets the bit alignment.

FALSE - Values are aligned to the MS bit and the LS bits are zero

TRUE - Values are aligned to the LS bit and the MS bits are zero



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



BitAlignent Sets the bit alignment.

FALSE - Values are aligned to the MS bit and the LS bits are zero

TRUE - Values are aligned to the LS bit and the MS bits are zero



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

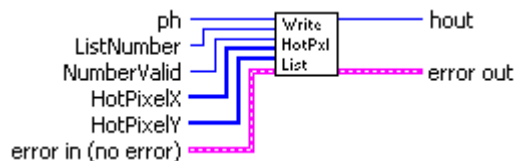


ErrorCode

4.6.5. WriteHotPixelList.vi

Creates or modifies a list of hot pixels inside the camera head. Check the availability of this feature in the camera description structure before using this function

Connector Pane



Front Panel

The front panel displays the following controls:

- ph:** A numeric control with a value of 0.
- hout:** A numeric control with a value of 0.
- ErrorCode:** A numeric control with a value of 0.
- ListNumber:** A numeric control with a value of 0.
- NumberValid:** A numeric control with a value of 0.
- HotPixelX:** A numeric control with a value of 0.
- HotPixelY:** A numeric control with a value of 0.
- error in (no error):** A status indicator showing a green checkmark and a value of 0.
- error out:** A status indicator showing a green checkmark and a value of 0.
- source:** A text area for inputting source information.

Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



ListNumber Identifies the list to modify by number (0 based)



NumberValid Number of hot pixels in the list. Must match the size of the HotPixelX and HotPixelY arrays.



HotPixelX Array of X coordinates of the hot pixels.



HotPixelY Array of Y coordinates of the hot pixels.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

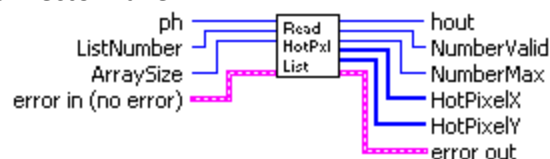


ErrorCode

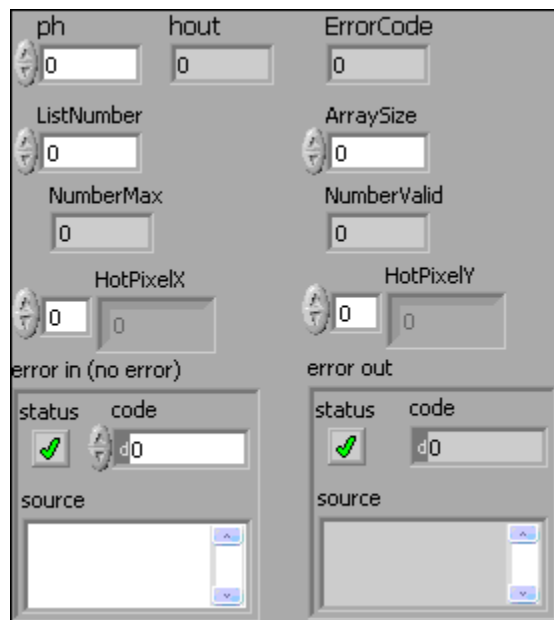
4.6.6. ReadHotPixelList.vi

Retrieves a list of hot pixels inside the camera head. Check the availability of this feature in the camera description structure before using this function












Connector Pane



Front Panel



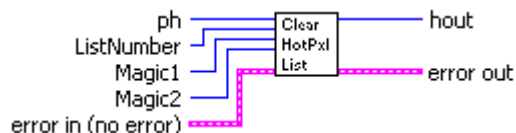
Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
-  **ListNumber** Identifies the list to read out by number (0 based)
-  **ArraySize** Size of the array used to hold the coordinates of the hot pixels.
-  **hout** Handle output
-  **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
-  **ErrorCode**
-  **NumberValid** Number of hot pixels in the list. Must match the size of the HotPixelX and HotPixelY arrays.
-  **HotPixelX** Array of X coordinates of the hot pixels.
-  **HotPixelY** Array of Y coordinates of the hot pixels.
-  **NumberMax** Shows the maximum possible number of hot pixels.

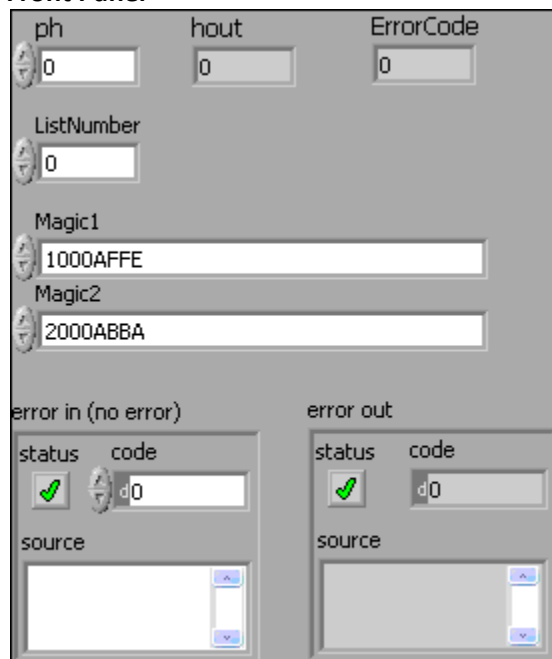
4.6.7. ClearHotPixelList.vi

Clears a list of hot pixels inside the camera head. Check the availability of this feature in the camera description structure before using this function. Two unlock codes must be given to prevent accidental deletion of the list.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



ListNumber Identifies the list to modify by number (0 based)



Magic1 First unlock code.



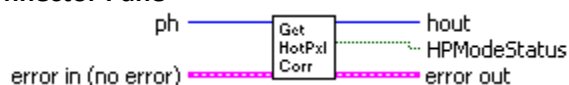
Magic2 Second unlock code.

**hout** Handle output**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs..**ErrorCode**

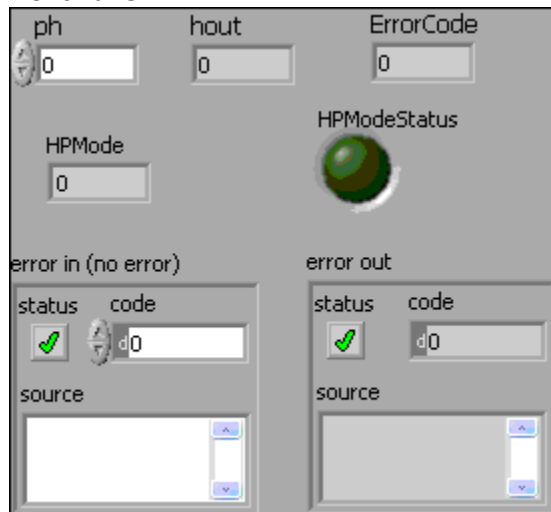
4.6.8. GetHotPixelCorrectionMode.vi

Finds the status of the hot pixel correction mode.

Connector Pane



Front Panel



Controls and Indicators

**ph** Handle for the camera**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs..**hout** Handle output**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs..

U32 **ErrorCode**

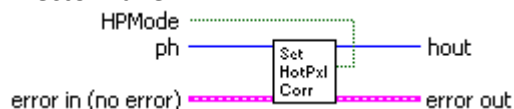
U16 **HPMode**

TF **HPModeStatus** TRUE if hot pixel correction is enabled.

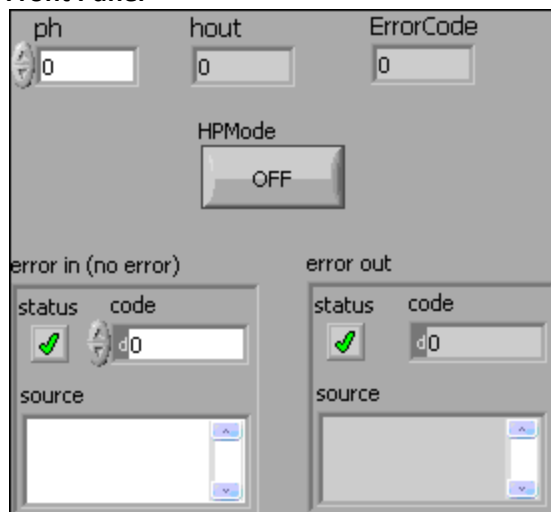
4.6.9.SetHotPixelCorrectionMode.vi

Turns the hot pixel correction mode on or off. Check the camera description to determine if hot pixel correction is supported

Connector Pane



Front Panel



Controls and Indicators

U64 **ph** Handle for the camera

TF **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs..

TF **HPMode** Turn hot pixel correction on or off.

U64 **hout** Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

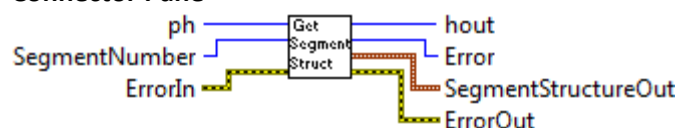


ErrorCode

4.6.10. GetSegmentStructure.vi

Provides details of the camRAM segment configuration, based on the ROI, binning and camRAM size, along with the partition information.

Connector Pane



Front Panel

The front panel of the GetSegmentStructure.vi interface includes the following elements:

- Controls:**
 - ph:** A numeric control with a value of 0.
 - SegmentNumber:** A numeric control with a value of 1.
 - ErrorIn:** An error cluster control with status (green checkmark), code (0), and source.
 - ErrorOut:** An error cluster control with status (green checkmark), code (0), and source.
- Indicators:**
 - SegmentStructureOut:** A large indicator showing a detailed configuration table.

Size	XRes	YRes	BinHor
60	60	60	60
BinVer	RODX0	ROIY0	ROIY1
60	60	60	60
Dummy1	ValidImageCnt	MaxImageCnt	
60	60	60	60
wDummy 1	wDummy 2	wDummy 3	wDummy 4
60	60	60	60
wDummy 5	wDummy 6	wDummy 7	wDummy 8
60	60	60	60
wDummy 9	wDummy 10	wDummy 11	wDummy 12
60	60	60	60
wDummy 13	wDummy 14	wDummy 15	wDummy 16
60	60	60	60
wDummy 17	wDummy 18	wDummy 19	wDummy 20
60	60	60	60
wDummy 21	wDummy 22	wDummy 23	wDummy 24
60	60	60	60
wDummy 25	wDummy 26	wDummy 27	wDummy 28
60	60	60	60
wDummy 29	wDummy 30	wDummy 31	wDummy 32
60	60	60	60
wDummy 33	wDummy 34	wDummy 35	wDummy 36
60	60	60	60
wDummy 37	wDummy 38	wDummy 39	wDummy 40
60	60	60	60

















Controls and Indicators



ph Handle to an open camera connection

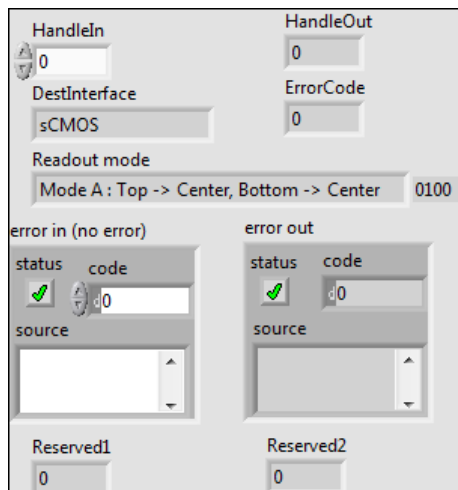
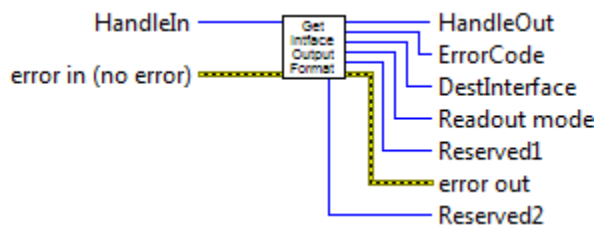


ErrorIn The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

-  **SegmentNumber** Segment number to inquire about.
-  **Error**
-  **hout** Handle to an open camera connection (same as input handle)
-  **ErrorOut** The **error in** cluster can accept error information wired from VIs previously called.
-  **SegmentStructureOut** Information about the selected segment, including ROI, number of valid images, and maximum number of images.
-  **Size**
 -  **XRes**
 -  **YRes**
 -  **BinHor**
 -  **BinVer**
 -  **ROI0**
 -  **ROI0**
 -  **ROI1**
 -  **ROI1**
 -  **ValidImageCnt**
 -  **MaxImageCnt**

4.6.11. GetInterfaceOutputFormat.vi

Displays options for the data interface to the pco.cameras. This VI can be used to query the readout modes on the pco.edge, and various output format options on the HDSDI interface of the pco.dimax.



U64 **HandleIn** Handle to selected camera

E **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called.

U16 **DestInterface** Selected interface type:

U16 **Reserved2** Reserved for future options

U16 **Reserved1** Reserved for future options

I32 **ErrorCode**

U64 **HandleOut** Handle for selected camera

E **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

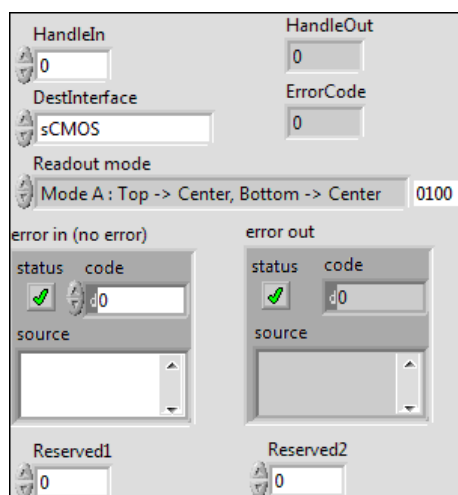
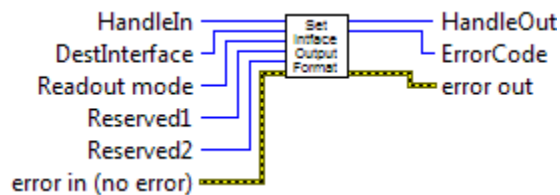
U16 **Readout mode** Data interface options

4.6.12. SetInterfaceOutputFormat.vi

Controls options for the data interface to the pco.cameras. This VI can be used to set the readout modes on the

File:	Version:	as of:	Author:	Page 213 of 252
MA_DCLABVIEWver204.doc	2.04	03.09.2015	MM/FRE/ LWA/ EO/ GHO	

pco.edge, and control various options on the HDSDI interface of the pco.dimax.



U16 **DestInterface** Select Interface type

U16 **Reserved2** Reserved for future options

U16 **Reserved1** Reserved for future options

U64 **HandleIn** Handle to selected camera

E16 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called.

U16 **Readout mode** Data interface options.

I32 **ErrorCode**

U64 **HandleOut** Handle for selected camera

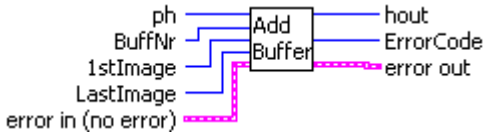
E16 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

4.7. APIManagement.llb

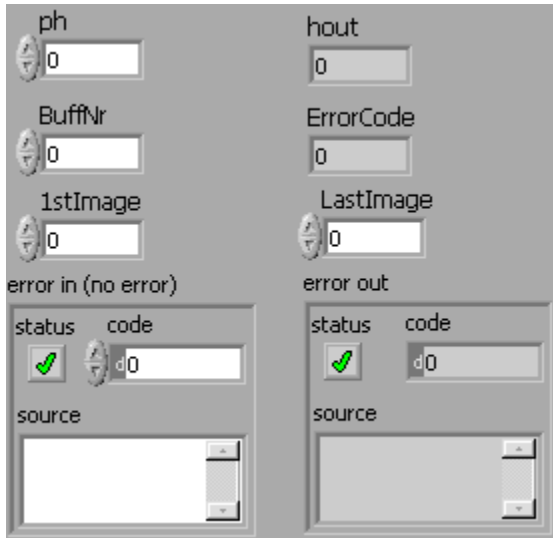
4.7.1.AddBuffer.vi (**Obsolete – Use AddBufferEx.vi for new development)

Adds a buffer to the driver queue. This function returns immediately. If the desired image is transferred to the buffer a buffer event will be created. Once an event is created, the data can be retrieved using GetImageBuffer.vi. This function is used to view images while the recording is enabled. To read out previously recorded images, use GetImage.vi.

Connector Pane



Front Panel



Controls and Indicators

U641 ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



1stImage Image to transfer to buffer.

0 - Transfers most recent image to the buffer

Any other number must be a valid image number. Value must also be the same as the LastImage



LastImage Image to transfer to buffer.

0 - Transfers most recent image to the buffer

Any other number must be a valid image number. Value must also be the same as the 1stImage



BuffNr Buffer to add to the queue.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

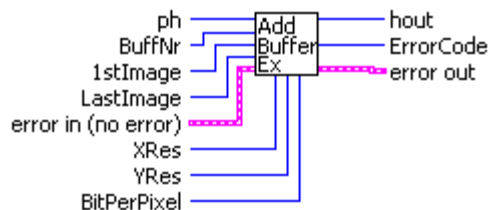


ErrorCode

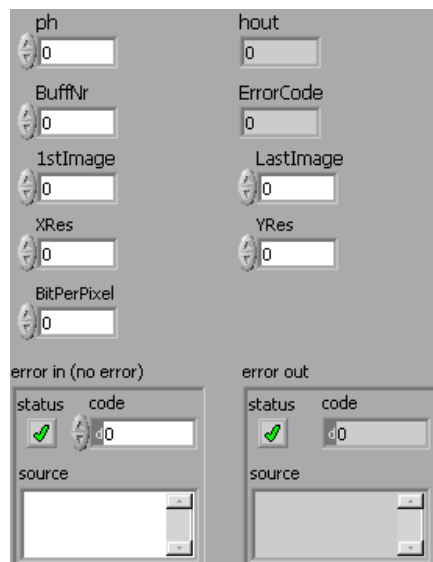
4.7.2. AddBufferEX.vi

Adds a buffer to the driver queue. This function returns immediately. If the desired image is transferred to the buffer a buffer event will be created. Once an event is created, the data can be retrieved using GetImageBuffer.vi. This function is used to view images while the recording is enabled. To read out previously recorded images, use GetImageEx.vi.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



1stImage Image to transfer to buffer.

0 - Transfers most recent image to the buffer

Any other number must be a valid image number. Value must also be the same as the LastImage



LastImage Image to transfer to buffer.

0 - Transfers most recent image to the buffer

Any other number must be a valid image number. Value must also be the same as the 1stImage



BuffNr Buffer to add to the queue.



XRes Horizontal resolution, in pixels, of the image to be transferred



YRes Vertical resolution, in pixels, of the image to be transferred



BitPerPixel Number of bits used to store one pixel of data. This parameter must match

the number specified in the CameraDescription.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

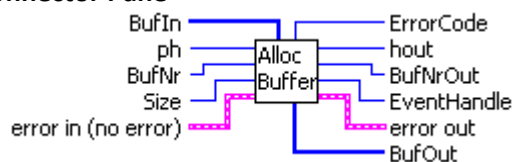


ErrorCode

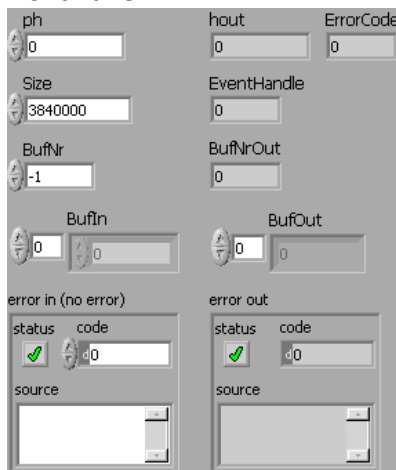
4.7.3.AllocateBuffer.vi

Allocates a buffer to receive the transferred images. There is a maximum of 8 buffers. This function is needed to create buffers for the image transfer. During recording you can get images with the AddBuffer function. While waiting for an image you can poll the buffer status with GetBufferStatus. Data can be retrieved using GetImageBuffer.vi once an event is detected.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed

in the event of errors from other VIs.

[I16] **BufNr** Number of the buffer to allocate. To allocate a new buffer, this parameter must be -1. To re-allocate an existing buffer, use that buffer use that buffer number.

[U32] **Size** Number of bytes to allocate for the buffer.

[U16] **BufIn** A LabVIEW array of unsigned 16 bit integers (U16), initialized to the exact size of the buffer. For example, for a buffer of 1600 X 1200 pixels, the array should have 1920000 elements.

[U64] **hout** Handle output

[F1] **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs..

[I32] **ErrorCode**

[U32] **EventHandle** Reserved for future versions

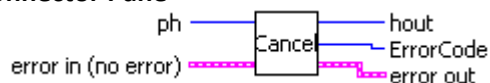
[I16] **BufNrOut** Number of the buffer now allocated

[U16] **BufOut** A LabVIEW array of unsigned 16 bit integers (U16), to hold the buffer data. The array will have as many elements as the pixels in the buffer.

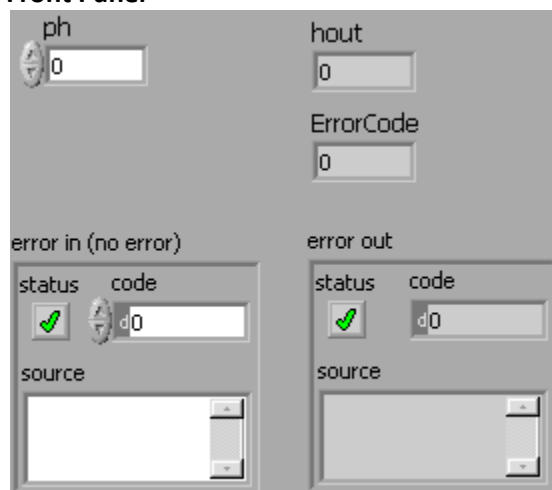
4.7.4. CancellImages.vi

Removes all buffers from the driver queue. Stops pending buffers while the camera is recording. Recording can then be terminated by setting the recording mode to "Stop". It is recommended that if there are pending buffers you should call CancellImages before you stop recording with SetRecordingState setting to zero.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

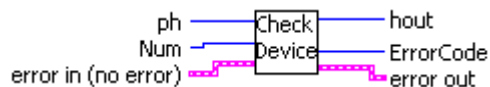


ErrorCode

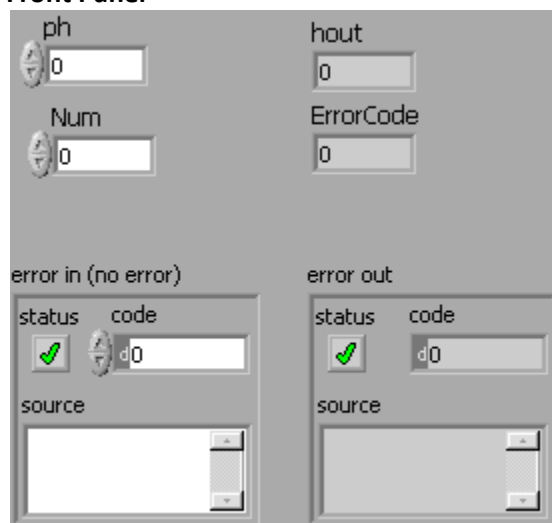
4.7.5. CheckDeviceAvailability.vi

This function can be used to determine if a device is still available after a bus reset. If the function returns without any errors, the device is still available.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



Num



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs..

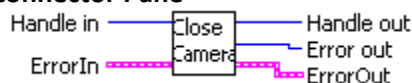


ErrorCode

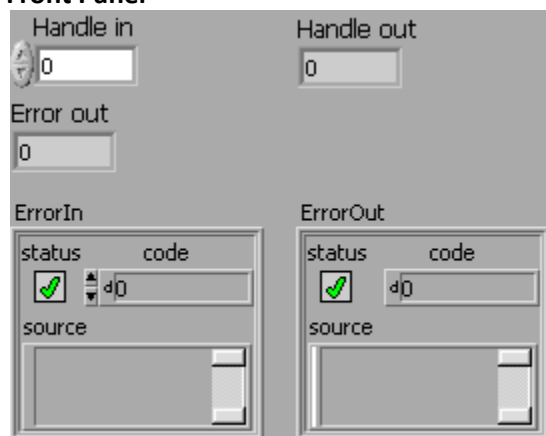
4.7.6. CloseCamera.vi

Closes a previously opened camera and returns resources to the operating system. It is strongly recommended to call this function before terminating the LabVIEW application.

Connector Pane



Front Panel



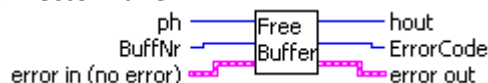
Controls and Indicators

- U32** **Handle in** Handle to a previously opened camera
- E51** **ErrorIn** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs..
- I32** **Error out**
- U32** **Handle out** Handle output
- E51** **ErrorOut** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

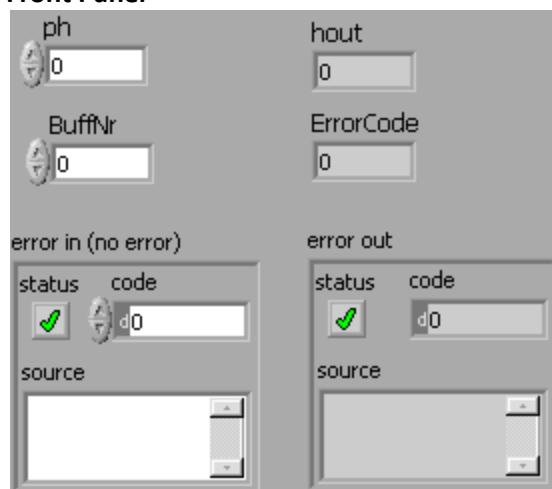
4.7.7.FreeBuffer.vi

Frees a previously allocated buffer. It is recommended to free all allocated buffers before the LabVIEW application terminates.







Connector Pane



Front Panel



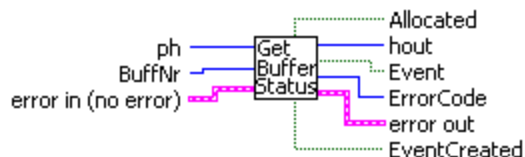
Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
-  **BuffNr** Buffer number to free.
-  **hout** Handle output
-  **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs..
-  **ErrorCode**

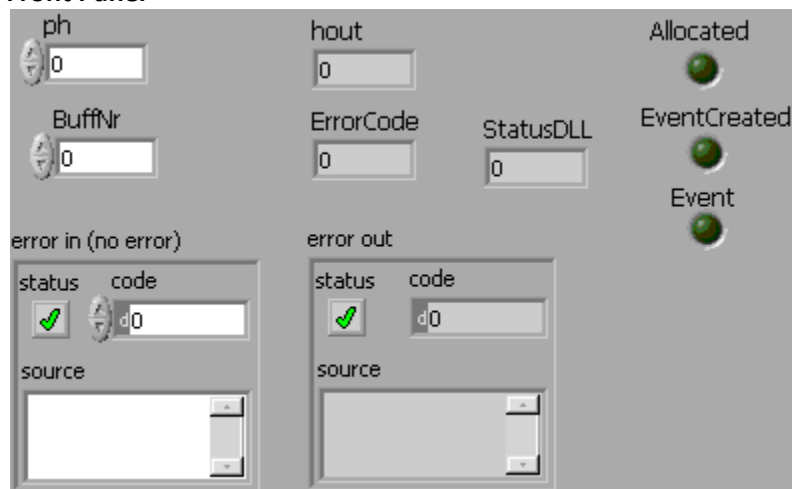
4.7.8. GetBufferStatus.vi

Get the buffer status of a previously 'allocated' and 'added' buffer. This can be used to poll the status, while waiting for an image during recording. The "event" flag will indicate when an image is ready for transfer.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



BuffNr Buffer to find the status of



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode

- TF

Event Indicates that an event has been generated and an image is available for transfer.
- TF

EventCreated Indicates that the buffer has been allocated and an event will be generated when an image is available.
- TF

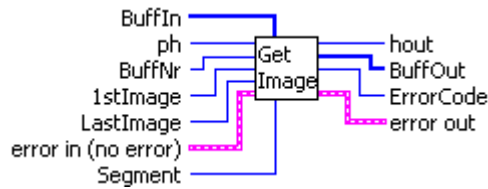
Allocated Indicates that the buffer has been allocated
- U32

StatusDLL Status word. See also the individual flags.

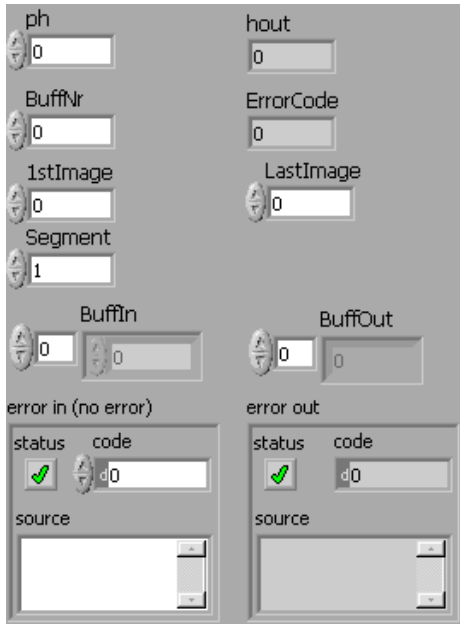
4.7.9. GetImage.vi (**Obsolete – Use GetImageEx.vi for new development**)

Gets previously recorded images from the camera. This function returns after the desired image is transferred to the buffer. You can get more than one image from the camera with this function call, but you have to take care about the size of the receiving buffer. To view images while the recording is enabled, use AddBuffer, and GetImageBuffer.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs..



1stImage Number of the first image to retrieve from the CamRam. This must be a valid image number. If attempting to retrieve more than one image, make sure that the buffer is of the correct size.



LastImage Number of the last image to retrieve from the CamRam. This must be a valid image number. If attempting to retrieve more than one image, make sure that the buffer is of the correct size.



BuffNr Number of a previously allocated buffer to receive the image data.



Segment Segment of camera memory to retrieve images from. Valid numbers are 1,2,3, and 4



BuffIn LabVIEW array to hold the image data. Array must be initialized to the correct size before use.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode

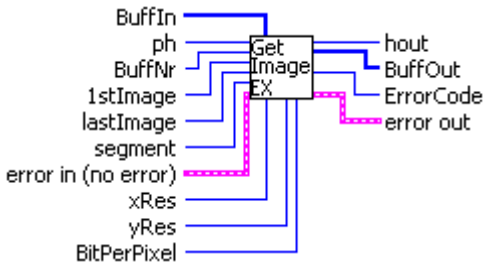


BuffOut LabVIEW array containing image data retrieved from the CamRam.

4.7.10. GetImageEX.vi

Gets an image from a previously allocated and added buffer, after an event is created.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs..



BuffNr Buffer number to get the image from



BuffIn LabVIEW array to hold the image data in the buffer. The array must be previously created with the exact number of elements required to hold the image.



xRes Horizontal Resolution of frame in buffer



yRes Vertical resolution of frame in buffer



BitPerPixel Pixel depth in bits (e.g. 14 bit, 12 bit)



LastImage Number of the last image to retrieve from the CamRam. This must be a valid image number. If attempting to retrieve more than one image, make sure that the buffer is of the correct size.



1stImage Number of the first image to retrieve from the CamRam. This must be a valid image number. If attempting to retrieve more than one image, make sure that the buffer is of the correct size.



segment Segment of camera memory to retrieve images from. Valid numbers are 1,2,3, and 4



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs..



ErrorCode

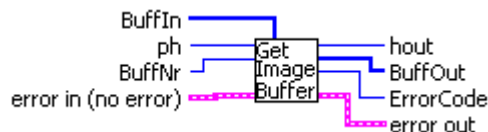


BuffOut LabVIEW array that contains the image data from the specified buffer.

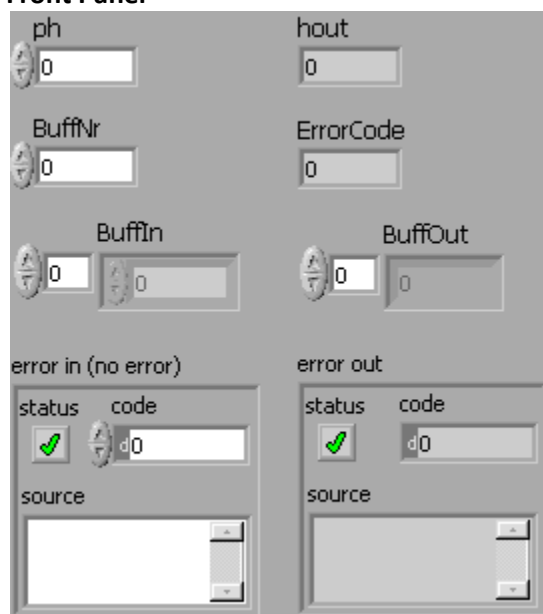
4.7.11. GetImageBuffer.vi

Gets an image from a previously allocated and added buffer, after an event is created.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs..



BuffNr Buffer number to get the image from



BuffIn LabVIEW array to hold the image data in the buffer. The array must be previously created with the exact number of elements required to hold the image.



hout Handle output

[E11] **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

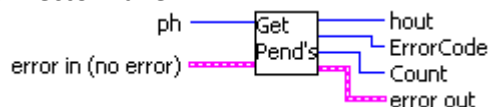
[I32] **ErrorCode**

[U16] **BuffOut** LabVIEW array that contains the image data from the specified buffer.

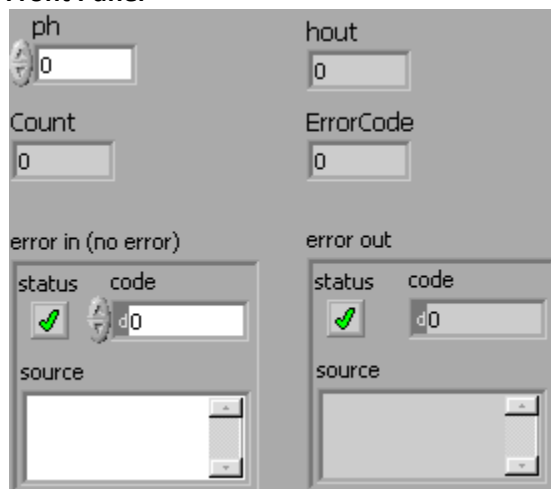
4.7.12. GetPendingBuffer.vi

Finds the number of buffers queued and ready to accept image data. This number should be found after stopping a recording. If there are buffers pending when the camera is stopped, they should be cleared using CancellImages.vi.

Connector Pane



Front Panel




Controls and Indicators

[U64] **ph** Handle for the camera

[E11] **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

[U64] **hout** Handle output

 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs..

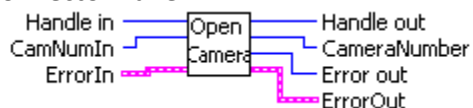
 **ErrorCode**

 **Count** Number of pending buffers.

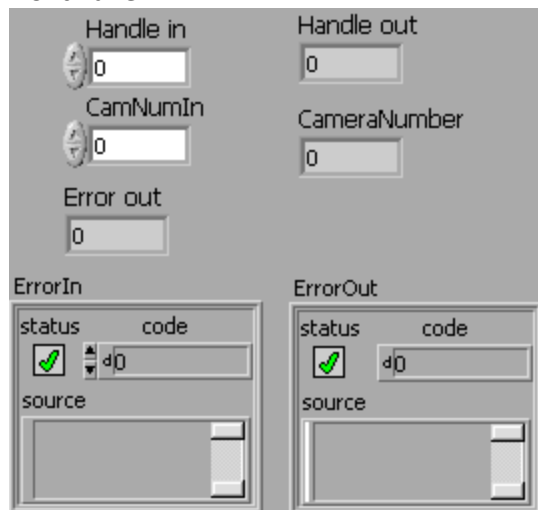
4.7.13. OpenCamera.vi

Opens a camera device. This VI must be called to initialize the camera before any other functions can be used.

Connector Pane





Front Panel



Controls and Indicators

 **Handle in** To start a new camera instance, the handle input must be 0.

 **CamNumIn** Number of the camera to be opened

 **ErrorIn** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs..

 **Error out**

- U32

Handle out A unique handle to communicate with the camera.
- U32

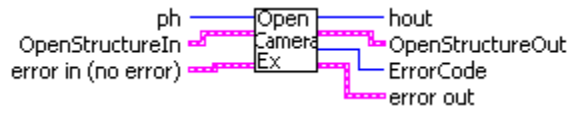
CameraNumber Camera number used in this "OpenCamera" command.
- E51

ErrorOut The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

4.7.14. OpenCameraEx.vi

Opens a camera device with given parameters, and returns a handle specific to that camera.

Connector Pane



Front Panel

The front panel of the OpenCameraEx.vi subVI. It features several input controls and two output indicator clusters. At the top, there are three numeric input fields: 'ph' (value 0), 'hout' (value 0), and 'ErrorCode' (value 0). Below these is the 'OpenStructureIn' cluster, which includes four numeric input fields: 'Size' (value 88), 'InterfaceType' (value 0), 'Camera Number' (value 0), and 'CameraNum AtInterface' (value 0). Below that are three more numeric input fields: 'OpenFlags' (value 0), 'OpenDWFlags' (value 0), and 'OpenPtrs' (value 0). The 'OpenStructureOut' cluster is located below the previous one, with the same four fields as 'OpenStructureIn', all set to 0. At the bottom, there are two error indicator clusters. The left one is labeled 'error in (no error)' and contains a 'status' indicator (green checkmark), a 'code' numeric input field (value 0), and a 'source' text area. The right one is labeled 'error out' and contains a 'status' indicator (green checkmark), a 'code' numeric input field (value 0), and a 'source' text area.

Controls and Indicators

- U64

ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



OpenStructureIn Configuration parameters for interfacing to pco.cameras



Size Size of the structure returned from the driver.



InterfaceType Describes the physical interface to the camera.

1 - FireWire/IEEE1394

2 - CameraLink using Matrox interface

3 - CameraLink using Silicon Software ME3 interface

4 - National Instruments 1400 series Camera Link boards

10 - CameraLink using generic serial interface

0xFFFF - Search for available interfaces

Other types may be added in future versions of the driver



Camera Number Requested number of the camera connected to this interface.



CameraNumAtInterface Assigned camera number for the selected interface. This may be different from the requested camera number, when scanning multiple connections for cameras.



OpenFlags Optional control flags for each interface. These are interface specific, and may be used to set certain parameters for that interface.



OpenFlags1



OpenDWFlags Additional optional control flags for each interface. These are interface specific, and may be used to set certain parameters for that interface.



OpenDWFlags1



OpenPtrs Additional interface data.



OpenPtr1



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs..



ErrorCode



OpenStructureOut Describes the parameters of the physical interface



Size Size of the structure returned from the driver.



InterfaceType Describes the physical interface to the camera.

1 - FireWire/IEEE1394

2 - CameraLink using Matrox interface

3 - CameraLink using Silicon Software ME3 interface

4 - National Instruments 1400 series Camera Link boards

10 - CameraLink using generic serial interface

0xFFFF - Search for available interfaces

Other types may be added in future versions of the driver



Camera Number Requested number of the camera connected to this interface.



CameraNumAtInterface Assigned camera number for the selected interface. This may be different from the requested camera number, when scanning multiple connections for cameras.



OpenFlags Optional control flags for each interface. These are interface specific, and may be used to set certain parameters for that interface.



OpenDWFlags Additional optional control flags for each interface. These are interface specific, and may be used to set certain parameters for that interface.

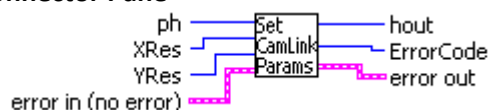


OpenPtrs Additional interface data.

4.7.15. CamLinkSetImageParameters.vi

Set the image parameters for the image buffer transfer inside the CamLink interface. When the pco.camera is connected to a CameraLink interface, this function must be called each time the size of the image is changed, before the user can retrieve images from the camera

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs..



XRes Actual x resolution of the image to be transferred



YRes Actual y resolution of the image to be transferred



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be

used by other VIs.

132 ErrorCode

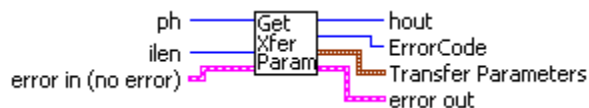
4.7.16. GetTransferParameters.vi

Finds the current transfer parameters of the communication interface. The parameters are interface-dependent, and must be interpreted accordingly.

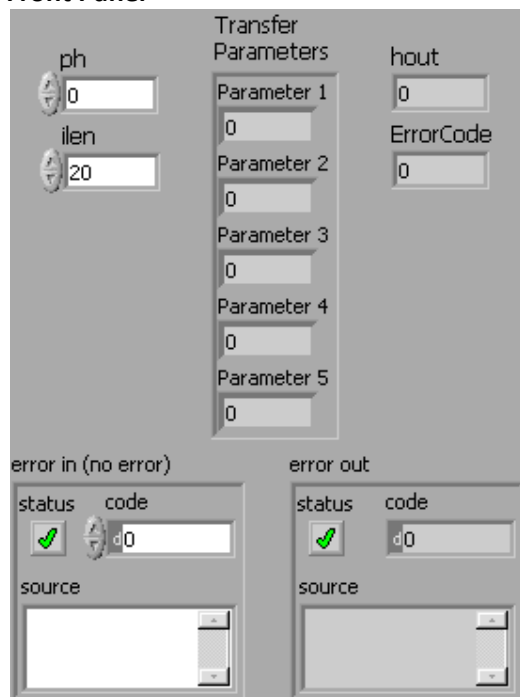
For FireWire cameras, the parameters describe the number of isochronous channels and bandwidth per channel for multiple cameras.

For CameraLink cameras, the parameters describe the baud rate for serial communications, as well as the function of the CC lines and the single/continuous transfer of images.

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



ilen Total number of bytes in "Transfer Parameters" cluster: default is 10.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



ErrorCode



Transfer Parameters Interface-specific parameters to control the transfer of data from camera to PC



Parameter 1 FireWire Bandwidth / CameraLink baud rate

Firewire cameras:

Bandwidth bytes - split between cameras. 4096 is default, use 2048 for 2 cameras, etc.

CameraLink:

Baud rate: Default is 9600. Use baud rates supported by your interface, typically 9600, 19200, 38400 etc.



Parameter 2 FireWire: Speed of iso transfer/ CamLink: Clock frequency

FireWire:

Finds speed of iso channel. Speed settings are 1 (slow), 2, 4 (fastest). Default value is 4 (recommended)

CameraLink:

Finds PixelClock. Should match the camera clock frequency. Values are in Hz: e.g. 40000000, 66000000, 80000000



Parameter 3 FireWire: Number of iso channels / CameraLink: CC Line settings

FireWire:

Determines the number of iso channels if more than one camera is connected.

CameraLink:

Displays the function of the CC lines in the CameraLink interface.

Bit0 set: CC1 line to be used as trigger instead of <exp trig>

Bit1 set: CC2 line to be used as acquire enable instead of <acq enbl>

Bit3 set: CC4 line to gate image transfer



Parameter 4 FireWire: Number of iso buffers / CameraLink: Data format

FireWire:

Sets number of iso buffers. Valid range is 16 to 256. Recommended value is 128

CameraLink:

Sets data format. Valid values are:

0x01: one pixel (16 bit) per clock

0x02: two pixels (12 bit) per clock (only for pco.hs1200 and not implemented yet)



Parameter 5 FireWire: Bytes per iso frame / CameraLink: Transmit enable.

FireWire:

Determines the number of bytes for each iso channel frame. Recommended value is 2000.

CameraLink:

Returns the current status of image transfer mode. Possible values:

0 - Single image transfer

1 - Continuous image transfers

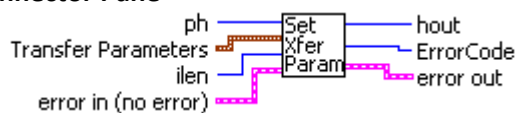
4.7.17. SetTransferParameters.vi

Controls the transfer parameters of the communication interface. The parameters are interface-dependent.

For FireWire cameras, the parameters control the number of isochronous channels and bandwidth per channel for multiple cameras.

For CameraLink cameras, the parameters set the baud rate for serial communications, as well as the function of the CC lines and the single/continuous transfer of images.

Connector Pane



Front Panel

The front panel of the SetTransferParameters.vi interface includes the following controls:

- ph**: A numeric control with a value of 0.
- ilen**: A numeric control with a value of 20.
- Transfer Parameters**: A group box containing five parameter controls, each with a value of 0:
 - Parameter 1
 - Parameter 2
 - Parameter 3
 - Parameter 4
 - Parameter 5
- hout**: A numeric control with a value of 0.
- ErrorCode**: A numeric control with a value of 0.
- error in (no error)**: A status section with a green checkmark icon, a "status" label, a "code" label with a value of 0, and a "source" label with a text area.
- error out**: A status section with a green checkmark icon, a "status" label, a "code" label with a value of 0, and a "source" label with a text area.

Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.



ilen Total number of bytes in "Transfer Parameters" cluster: default is 20.



Transfer Parameters Interface-specific parameters to control the transfer of data from camera to PC



Parameter 1 FireWire Bandwidth / CameraLink baud rate

Firewire cameras:

Bandwidth bytes - split between cameras. 4096 is default, use 2048 for 2 cameras, etc.

CameraLink:

Baud rate: Default is 9600. Use baud rates supported by your interface, typically 9600, 19200, 38400 etc.



Parameter 2 FireWire: Speed of iso transfer/ CamLink: Clock frequency

FireWire:

Sets speed of iso channel. Speed settings are 1 (slow), 2, 4(fastest). Default value is 4 (recommended)

CameraLink:

Sets PixelClock. Should match the camera clock frequency. Values are in Hz: 40000000, 66000000, 80000000



Parameter 3 FireWire: Number of iso channels / CameraLink: CC Line settings

FireWire:

Use this parameter to set the number of iso channels if more than one camera is connected. Use -1 to detect the number of channels automatically.

CameraLink:

Set the function of the CC lines in the CameraLink interface.

Bit0 set: enable CC1 line to be used as trigger instead of <exp trig>

Bit1 set: enable CC2 line to be used as acquire enable instead of <acq enbl>

Bit3 set: enable CC4 line to gate image transfer



Parameter 4 FireWire: Number of iso buffers / CameraLink: Data format

FireWire:

Sets number of iso buffers. Valid range is 16 to 256. Recommended value is 128

CameraLink:

Sets data format. Valid values are:

0x01: one pixel (16 bit) per clock

0x02: two pixels (12 bit) per clock (only for pco.hs1200 and not implemented yet)



Parameter 5 FireWire: Bytes per iso frame / CameraLink: Transmit enable.

FireWire:

Sets number of bytes for each iso channel frame. Recommended value is 2000

CameraLink:

Enables continuous transfer of images. Possible values:

0 - Single image transfer

1 - Continuous image transfers



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.



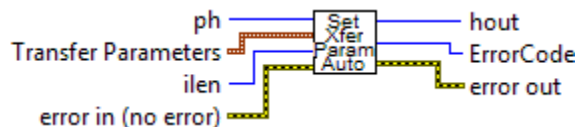
ErrorCode

4.7.18. SetTransferParametersAuto.vi

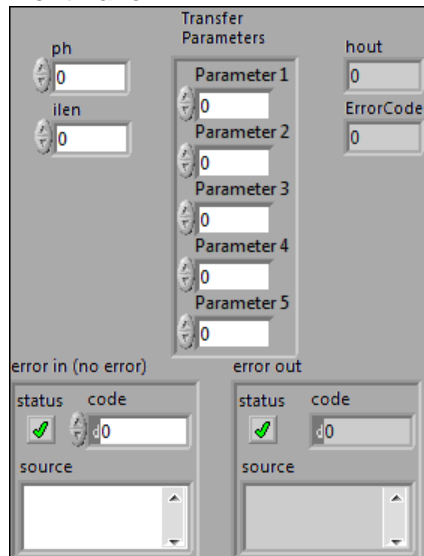
Automatically sets the transfer parameters for a pco.edge 5.5. This is the recommended function when Soft-ROI is enabled. This function replaces G(S)etTransferParameter.vi and SetActiveLookupTable.vi.

File:	Version:	as of:	Author:	Page 241 of 252
MA_DCLABVIEWver204.doc	2.04	03.09.2015	MM/FRE/ LWA/ EO/ GHO	








Connector Pane



Front Panel



Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called.
-  **ilen** Total number of bytes in "Transfer Parameters" cluster: default is 0.
-  **Transfer Parameters** Interface-specific parameters to control the transfer of data from camera to PC. This can be left uninitialized to set the parameters automatically.
-  **hout** Handle output
-  **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
-  **ErrorCode**

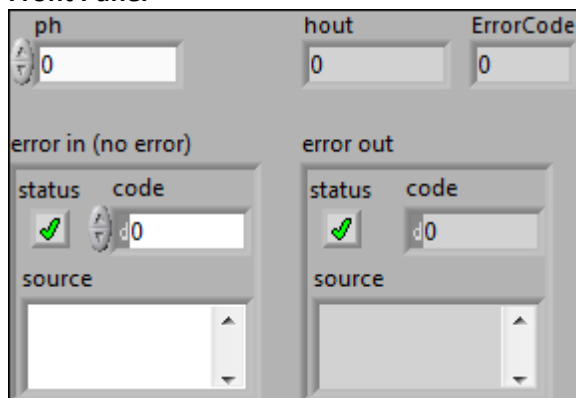
4.7.19. Cleanup.vi

Cleans up after the pcolabview.dll

Connector Pane



Front Panel



Controls and Indicators



error in (no error) The **error in** cluster can accept error information wired from VIs previously called.



ph Handle for the camera



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs

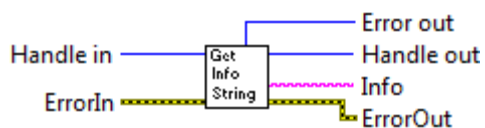


hout Handle output

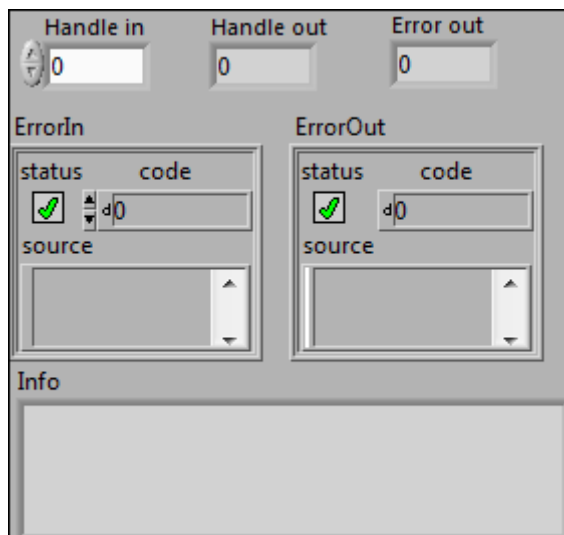
4.7.20. GetInfoString.vi

Retrieve version information etc. stored in camera head

Connector Pane



Front Panel



Controls and Indicators



Handle in Handle to a connected camera



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.



Error out Error number from camera driver



Handle out Handle of the connected camera



ErrorOut The **error in** cluster can accept error information wired from VIs previously called.

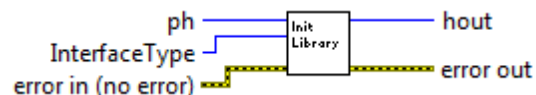


Info Information on the connected camera, including the camera type, interface type and camera number.

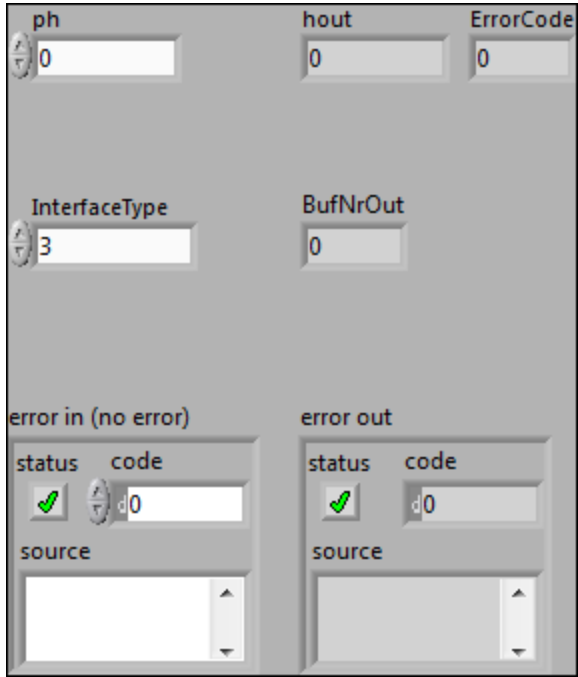
4.7.21. InitLibrary.vi

Loads the dynamic libraries for interfacing LabVIEW to the camera driver.






Connector Pane



Front Panel



Controls and Indicators

-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called.
-  **InterfaceType** Initializes the library for different camera types. Use 3 for pc^o.camera.
-  **ph** Handle for the camera
-  **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
-  **hout** Handle output

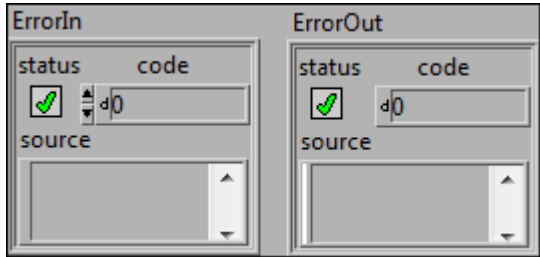
4.7.22. **Reset_SDK_Lib.vi**

Re-initialize interface library if an error occurs.

Connector Pane



Front Panel



Controls and Indicators



ErrorIn The **error in** cluster can accept error information wired from VIs previously called.

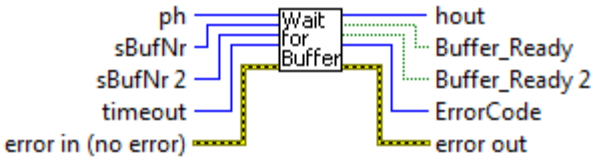


ErrorOut The **error in** cluster can accept error information wired from VIs previously called.

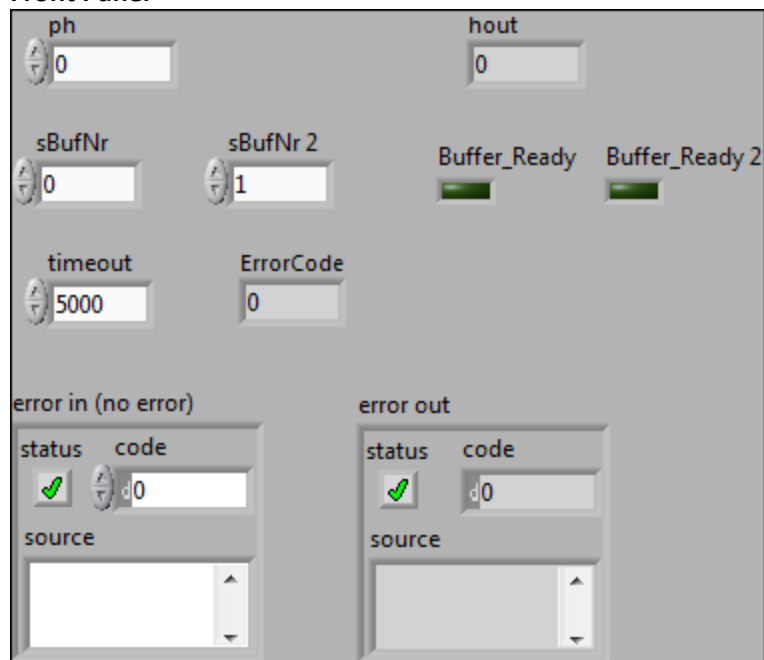
4.7.23. WaitForBuffer.vi

Waits for a buffer event, then returns a flag signaling which of two buffers has the event.











Connector Pane



Front Panel



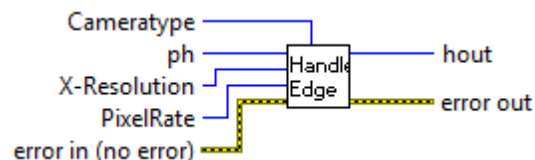
Controls and Indicators

-  **ph** Handle for the camera
-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously
-  **timeout** Maximum time to wait in milliseconds for a buffer event.
-  **sBufNr** Buffer number for one of two buffers in queue.
-  **sBufNr 2** Buffer number for another of two buffers in queue.
-  **hout** Handle output
-  **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.
-  **ErrorCode**
-  **Buffer_Ready** Signals buffer 1 is ready
-  **Buffer_Ready 2** Signals buffer 2 is ready

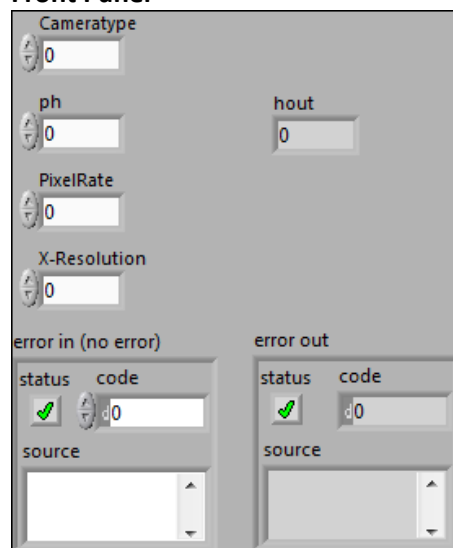
4.7.24. Handle_Edge_Pixelrate.vi

Turns pco.edge LUT operation on or off, based on the pixel rate and the horizontal resolution. If the clock rate is 286 MHz and the horizontal resolution is greater than 1920 pixels, the CameraLink interface cannot move the data fast enough in 16-bit per pixel mode. LUT's are used to reduce the pixel depth to 12 bits or less. This VI can be used to automatically enable or disable the LUT functions.


Connector Pane




Front Panel




Controls and Indicators

 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called.

 **ph** Handle for the camera

 **X-Resolution** X resolution as selected by the current ROI parameters, effectively (ROIX1 - ROIX0)

 **PixelRate** Sets the pixel rate. Takes effect at next ArmCamera command. The pixelrate is specified in Hz, e.g. 10000000 = 10 Mhz. Use GetDescription.vi to find the allowed rates for the camera.

 **CameraType** An integer that identifies the model of camera connected



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

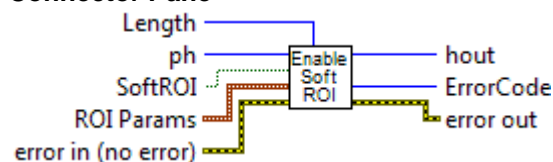


hout Handle output

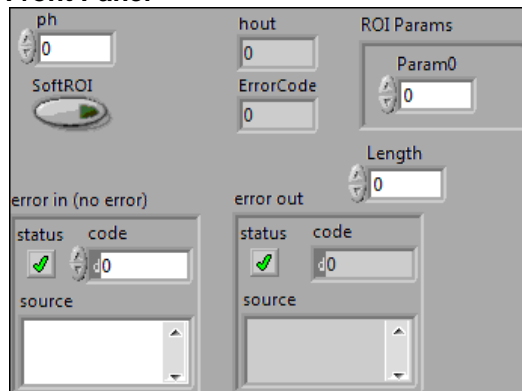
4.7.25. EnableSoftROIMode.vi

Enables Soft-ROI functionality for Soft-ROI capable interfaces. If it is necessary to get a smaller ROI-granularity (e.g. in x-direction it is only possible to set the ROI in steps of 160 pixels with a pco.edge 5.5) this function enables smaller granularity (e.g. a pco.edge 5.5 is reduced to 4 pixels in x-direction). If Soft-ROI is enabled it is recommended to use SetTransferParametersAuto.vi. This makes sure that the camera and interface are set to the correct transfer modes for Soft-ROI. GetTransferParameter.vi, SetTransferParameter.vi and SetActiveLookupTable.vi are replaced by SetTransferParametersAuto.vi. If SetTransferParametersAuto.vi is not used it is mandatory to ensure the correct setup of the transfer parameters (e.g. Soft-ROI is smaller than x=1920, but the camera ROI is bigger than x=1920 due to the granularity of the camera).

Connector Pane



Front Panel



Controls and Indicators



ph Handle for the camera



error in (no error) The **error in** cluster can accept error information wired from VIs previously called



SoftROI Enables or disables Soft ROI mode in cameras that have this capability.

FALSE - Disable Soft ROI mode

TRUE - Enable Soft ROI mode.



Length Length of the Soft ROI Params structure. Set to 0.



ROI Params ROI parameters for Soft ROI mode. Not currently implemented.



hout Handle output



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ErrorCode

5. Error / Warning Codes

The error codes are standardized where possible. The error codes contain the error layer information, the source (microcontrollers, CPLDs, FPGAs) and an error code (error cause). All values are combined by a logical OR operation. Error codes and warnings are always negative values, if read as signed integers, or if read as unsigned word, the MSB is set. Errors have the general format 0x80#####; warnings have the format 0xC0#####.

The error numbers are not unique. Each layer and the common errors have their own error codes. The “Error.vi” formats the error code into LabVIEW clusters, where the error text gives the layer and source information and descriptive text.



PCO AG

Donaupark 11

D-93309 Kelheim

fon +49 (0)9441 2005 0

fax +49 (0)9441 2005 20

eMail: info@pco.de

www.pco.de

PCO-TECH Inc.

6930 Metroplex Drive,

Romulus, Michigan 48174, USA

www.pco-tech.com