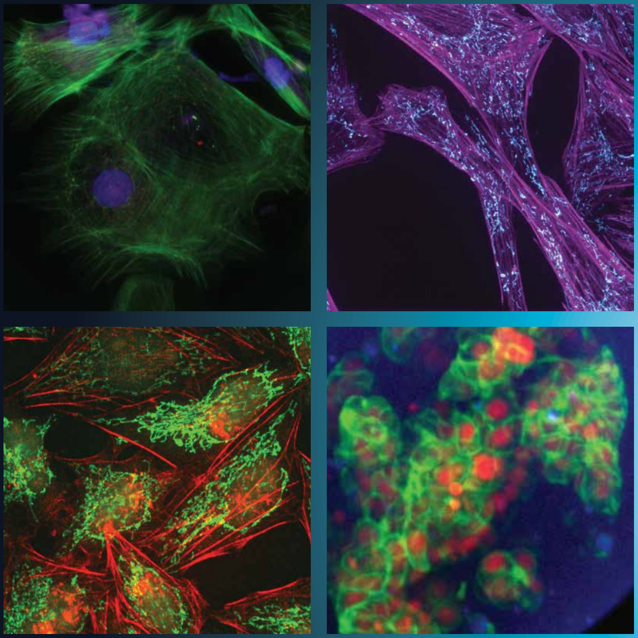
PHOTOMETRICS®

Scientific CMOS, EMCCD and CCD Cameras



PVCamNET with LabVIEW

USER MANUAL

PHOTOMETRICS®
Scientific CMOS, EMCCD and CCD Cameras

© Copyright 2018 Photometrics
3440 East Britannia Drive, Suite 100
Tucson, Arizona 85706
Tel: +1 520.889.9933
Fax: +1 520.295.0299

The information in this publication is believed to be accurate as of the publication release date. However, Photometrics does not assume any responsibility for any consequences including any damages resulting from the use thereof. The information contained herein is subject to change without notice. Revision of this publication may be issued to incorporate such change.

Table of Contents

Chapter 1.

OVERVIEW

This document provides LabVIEW programmers with an overview of using the PVCamNET Assembly to write their own virtual instruments (VIs). Included with this document are a demo application example VI and simple example VIs.

The demo application is event driven to illustrate nearly all the supported features from PVCamNET. The simple examples demonstrate the order of PVCamNET function calls needed to properly initialize the PVCAM library, open a camera, capture an image, close the camera, and finally uninitializing the PVCAM library.

VI sample code is explained on its VI using screenshots or on a recorded video clip that can be viewed.

The getting started section talks about the property nodes of the PVCamNET Assembly and how to use the interface. In the quick start guide, it explains how to start writing VI code using PVCamNET.

Chapter 2.

INTRODUCTION

System Requirements

1. Compatible cameras
   a. Prime 95B, and Prime BSI
   b. Retiga R1, R3 and R6 (Mono and Color)
   c. Iris 9 and 15
   d. Other currently available cameras using PVCAM should also work but have not been tested for incompatibilities.
2. Software
   a. PVCAM version:3.8.0.6 or later
   b. PVCAM library version: 8.0.1218.0
   c. Device driver version: 2.0.0
   d. PVCamNET version: 1.0.126 or later
   e. LabVIEW 2018 Version 18.0f2 (64-bit) or later.
3. Computer
   a. DDR 8GB minimum (16GB recommended)

Known Issues

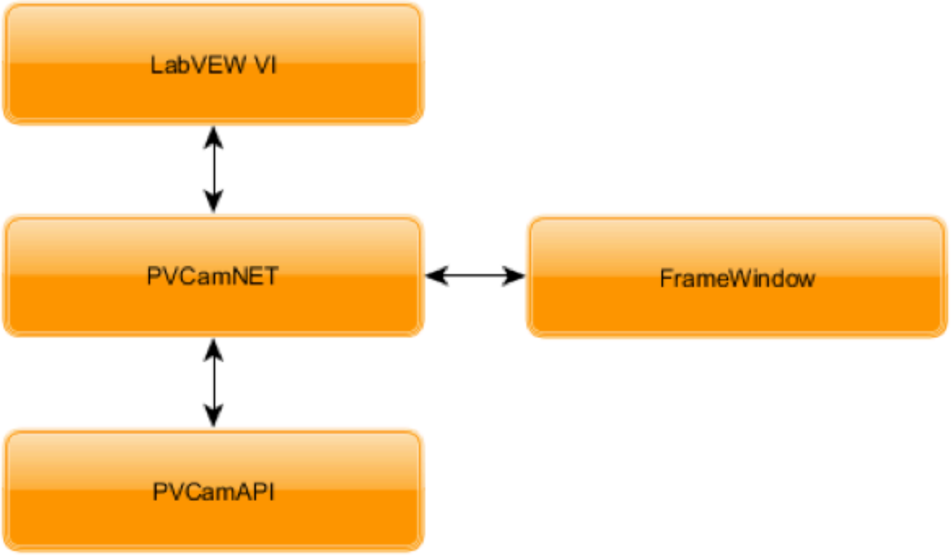
1. LabVIEW's crash reporter messages may appear after manually exiting LabVIEW.
2. LabVIEW may crash if the user tries unsupported cameras.
3. The user must not click "Abort execution" or "Run continuously". If either of the buttons is clicked, then the LabVIEW application must be restarted.
4. Opening FanSpeedControl.vi with an Iris camera may throw an exception. Hit continue to proceed.
5. When saving images if multiple PMQI's LabVIEW projects are opened saved images are found at the first PMQI project's location.
6. When running the application with a larger sensor camera such as Iris15 a large chunk of memory may not be freed up immediately. After a while the garbage collector is invoked to deallocate the memory used by the application.
7. SequenceAcquisitionSaveMultipleTiffImage.vi may not save all frames.
8. Frames in buffer may skip order if the computer's ram size is smaller than 8GB.
9. Some cameras may not show histogram as expected
10. Framerate may not be the same as the framerate on the datasheet.

Limitations

1. Post processing features such as “PrimeLocate” are not supported in the current release
2. Cameras must be turned on, connected to the computer, and recognized by the host operating system before launching LabVIEW.
3. The user must use the VIs and PVCamNET.dll enclosed in the same package.
4. The user cannot open multiple cameras simultaneously.
5. The user cannot run multiple VIs simultaneously.

Plugin Architecture

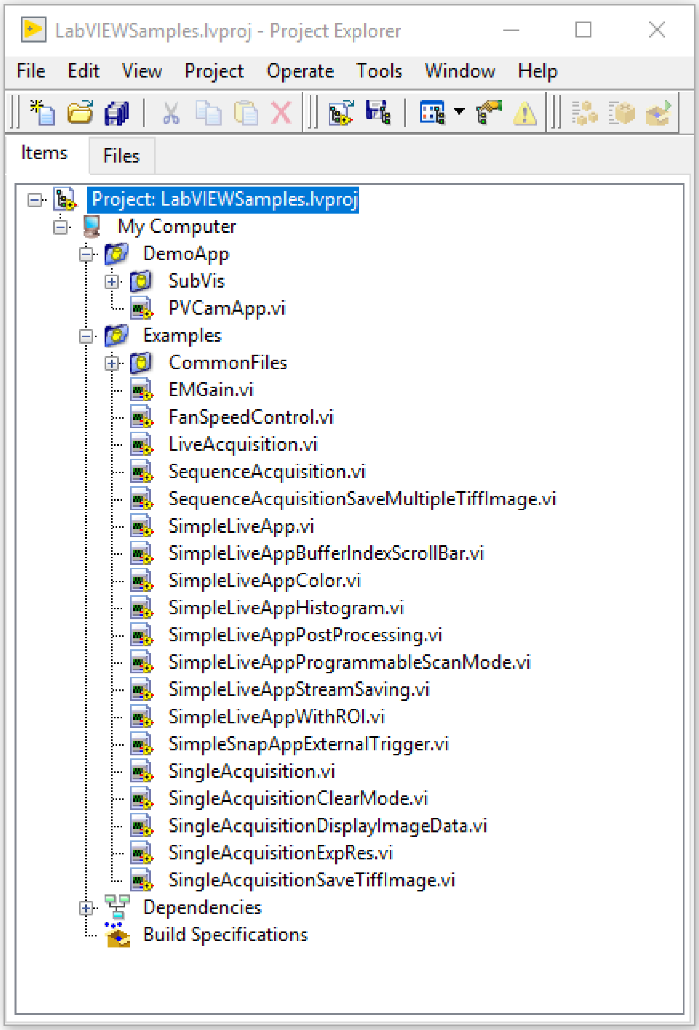
LabVIEW gives the ability to use .NET assemblies through .NET nodes. PVCamNET is a .NET assembly which allows the user to use its nodes for virtual instruments such as “PVCamApp.vi” and “SingleAcquisition.vi” to control PVCAM cameras. From the user’s perspective, there are two graphical user interfaces; one allows the user to control the camera (LabVIEW VI’s front panel) and the other displays acquired images (FrameWindow). To control a camera, the user must open one of the provided VIs (“PVCamApp.vi”, “SingleAcquisition.vi”, and so on). Once the camera is open with a selected VI, FrameWindow is generated by the PVCamNET assembly to display images.



Note that a more detailed description of the PVCamNET assembly is available as part of PVCamNET distribution package.

Plugin Structure

The plugin consists of a demo VI and example VIs shown below.



LabVIEWSamples.lvproj - Project Explorer

```
Project: LabVIEWSamples.lvproj
  My Computer
    DemoApp
      SubVis
      PVCamApp.vi
    Examples
      CommonFiles
      EMGain.vi
      FanSpeedControl.vi
      LiveAcquisition.vi
      SequenceAcquisition.vi
      SequenceAcquisitionSaveMultipleTiffImage.vi
      SimpleLiveApp.vi
      SimpleLiveAppBufferIndexScrollBar.vi
      SimpleLiveAppColor.vi
      SimpleLiveAppHistogram.vi
      SimpleLiveAppPostProcessing.vi
      SimpleLiveAppProgrammableScanMode.vi
      SimpleLiveAppStreamSaving.vi
      SimpleLiveAppWithROI.vi
      SimpleSnapAppExternalTrigger.vi
      SingleAcquisition.vi
      SingleAcquisitionClearMode.vi
      SingleAcquisitionDisplayImageData.vi
      SingleAcquisitionExpRes.vi
      SingleAcquisitionSaveTiffImage.vi
    Dependencies
    Build Specifications
```

Chapter 2.

DEMONSTRATION

Demo App

PVCamAPP.vi

- An event driven app to control the basic features of a PVCam camera
- Supports setting readout ports, speeds, gain, clear cycle, clear mode, exp.out, trigger, exposure time, exposure resolution, binning, ROI, snap, live mode, histogram, and saving an image.

SubVis/PopulateClearCycles.vi

- A sub vi retrieves supported clear cycles from the camera and populates on the GUI.

SubVis/PopulateClearModes.vi

- A sub vi retrieves supported clear modes from the camera and populates on the GUI.

SubVis/PopulateColorMode.vi

- A sub vi retrieves supported color mode from the camera and populates on the GUI.

SubVis/PopulateExposureResolution.vi

- A sub vi retrieves supported exposure resolution from the camera and populates on the GUI.

SubVis/PopulateExposureTime.vi

- A sub vi retrieves supported exposure time from the camera and populates on the GUI.

SubVis/PopulateExpOut.vi

- A sub vi retrieves supported exposure out from the camera and populates on the GUI.

SubVis/PopulatePortsAndSpeedsGain.vi

- A sub vi retrieves supported ports and speeds from the camera and populates on the GUI.

SubVis/PopulatePostProcessing.vi

- A sub vi retrieves supported post proessing from the camera and populates on the GUI.

SubVis/PopulateSupportedBinning.vi

- A sub vi retrieves supported binning from the camera and populates on the GUI.

SubVis/PopulateTriggerModes.vi

- A sub vi retrieves supported trigger modes from the camera and populates on the GUI.

SubVis/PVCamGlobalVariables.vi

- Global variables used by PVCamApp.vi

SubVis/PVCamPropertyChangedCallbackForCameraState.vi

- The callback vi invoked by PVCamNET used by PVCamAPP.vi for the camera state

SubVis/PVCamPropertyChangedCallbackForHistogram.vi

- The callback vi invoked by PVCamNET used by PVCamAPP.vi for the histogram

SubVis/PVCamPropertyChangedEventCallbackForBufferIndexDroppedFrame.vi

- The callback vi invoked by PVCamNET used by PVCamAPP.vi for the buffer index, dropped frame and scroll bar

SubVis/SetBinning.vi

- A sub vi sets binning.

SubVis/SetClearCycles.vi

- A sub vi sets clear cycles.

SubVis/SetClearingMode.vi

- A sub vi sets clearing mode.

SubVis/SetColorMode.vi

- A sub vi sets color mode.

SubVis/SetExposureResolution.vi

- A sub vi sets exposure resolution.

SubVis/SetExposureTime.vi

- A sub vi sets exposure time.

SubVis/SetExpOut.vi

- A sub vi sets exp out.

SubVis/SetGain.vi

- A sub vi sets gain.

SubVis/SetPortsAndSpeedsGain.vi

- A sub vi sets ports and speeds gain.

SubVis/SetPostProcessing.vi

- A sub vi sets post processing feature.

SubVis/SetTriggerMode.vi

- A sub vi sets trigger mode.

Examples

EMGain.vi

- An EM gain example

FanSpeedControl.vi

- A fan speed control example

LiveAcquisition.vi

- A live mode example

SequenceAcquisition.vi

- A lat sequence example

SequenceAcquisitionSaveMultipleTiffImage.vi

- A flat sequence example demonstrates saving a TIFF image stack

SimpleLiveApp.vi

- An event driven live mode app

SimpleLiveAppBufferIndexScrollBar.vi

- An event driven live mode app demonstrates updating the camera window with a scroll bar

SimpleLiveAppColor.vi

- An event driven live mode app demonstrates color mode

SimpleLiveAppHistogram.vi

- An event driven live mode app demonstrates a histogram

SimpleLiveAppPostProcessing.vi

- An event driven live mode app demonstrates post processing features.

SimpleLiveAppProgrammableScanMode.vi

- An event driven live mode app demonstrates programmable scan mode features

SimpleLiveAppStreamSaving.vi

- An event driven live mode app demonstrates stream saving

SimpleLiveAppWithROI.vi

- An event driven live mode app demonstrates ROI

SimpleSnapAppExternalTrigger.vi

- An event driven snap mode app demonstrates external trigger

SingleAcquisition.vi

- A single acquisition example

SingleAcquisitionClearMode.vi

- A single acquisition example demonstrates clear mode

SingleAcquisitionDisplayImageData.vi

- A single acquisition example demonstrates displaying image data

SingleAcquisitionExpRes.vi

- A single acquisition example demonstrates updating the exposure resolution

SingleAcqusitionSaveTiffImage.vi

- A flat sequence single acquisition example saves acquired image to a TIFF file

CommonFiles/GlobalVariables.vi

- Global variables used by VIs used under Examples

CommonFiles/PropertyChangedEventCallbackForBufferIndex.vi

- The callback vi invoked by PVCamNET for the buffer index

CommonFiles/PropertyChangedEventCallbackForCameraState.vi

- The callback vi invoked by PVCamNET for the camera state

CommonFiles/PropertyChangedEventCallbackForCapturedFrames.vi

- The callback vi invoked by PVCamNET for the captured frames

CommonFiles/PropertyChangedEventCallbackForHistogram.vi

- The callback vi invoked by PVCamNET for the histogram

CommonFiles/PropertyChangedEventCallbackForScrollbar.vi

- The callback vi invoked by PVCamNET for the scrollbar

Dependencies

mscorlib

- The library contains GC::Collect()

PVCamNET.dll

- The PVCamNET.dll used to communicate with PVCam cameras

System

- The system library for LabVIEW

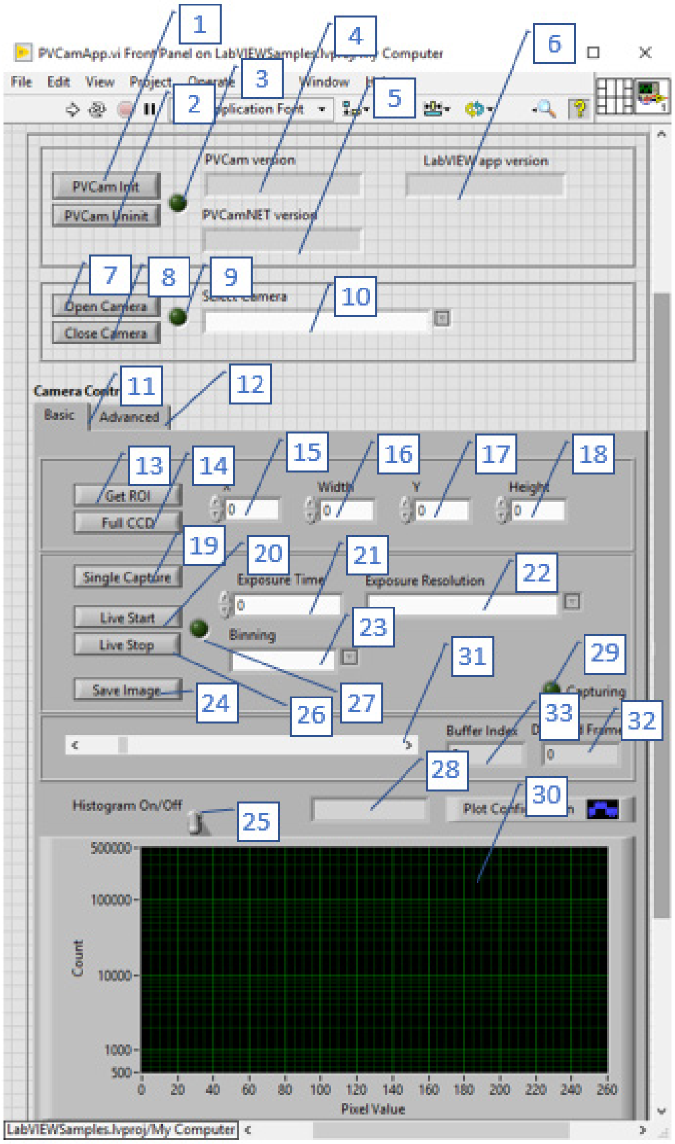
Chapter 3.

USING THE PLUGIN

Plugin Installation

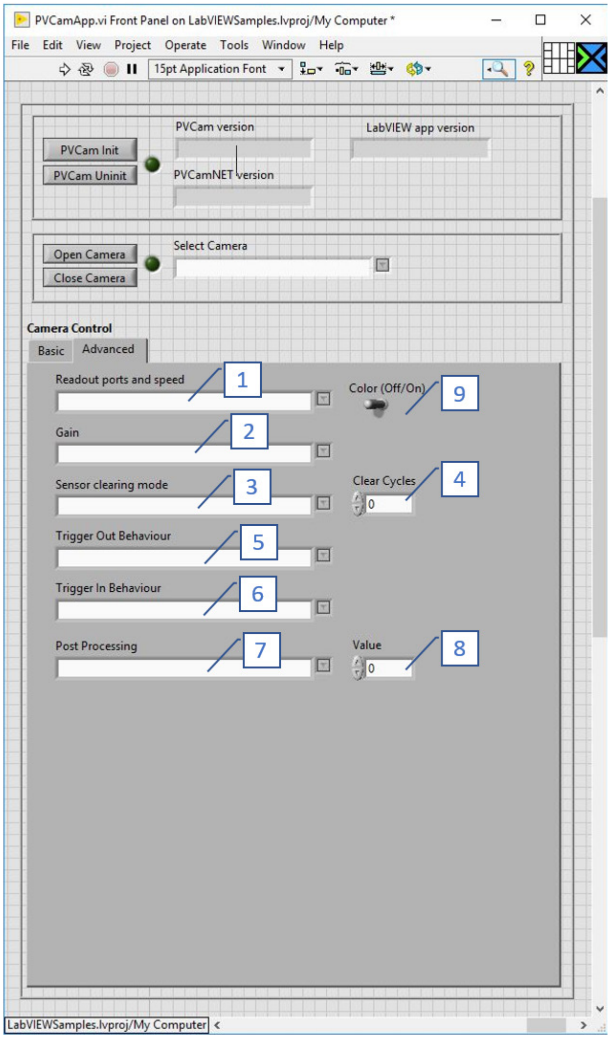
1. Download and install the PVCAM runtime from the Photometrics website https://www.photometrics.com
2. Download and install the LabVIEW version specified in the requirement section above
3. Connect a camera specified in the requirement section above
4. Open the LabVIEW project (LabVIEWSamples.lvproj)

PVCamApp.vi



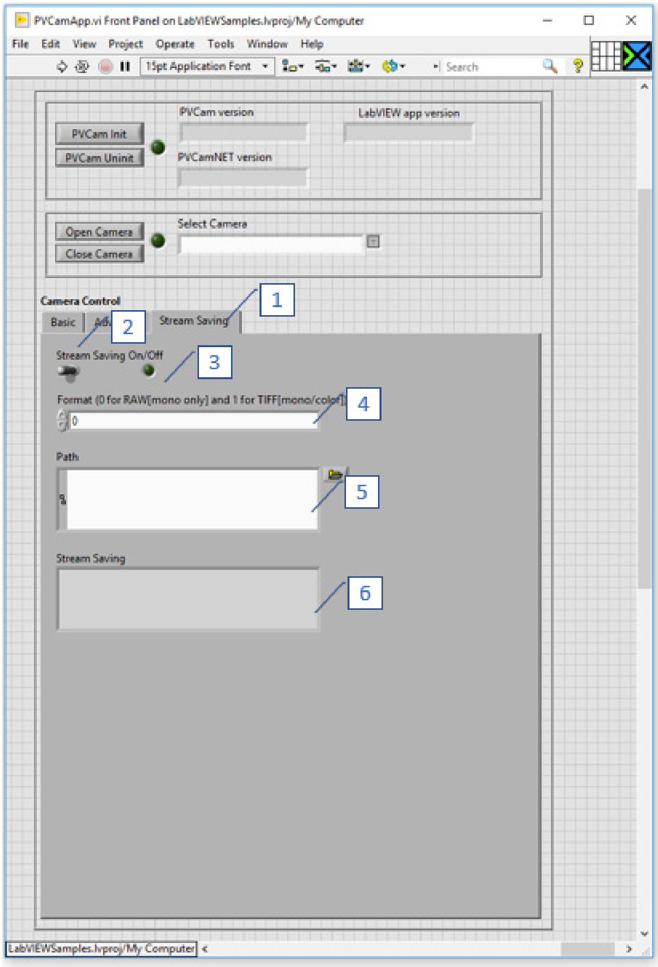
UI layout

1. PVCam Init button initializes the PVCamNET library
2. PVCam Uninit button uninitializes the library
3. PVCam status indicator turns on if the library is successfully initialized and otherwise is off
4. PVCam version displays the version of PVCam upon successful initialization of the library
5. PVCamNET version displays the version of PVCamNET
6. LabVIEW app version displays the version of DemoApp.vi upon calling DemoApp.vi
7. Open Camera button allows to open the selected camera otherwise is off
8. Close Camera button closes the camera
9. Camera Status indicator turns on if the camera is open
10. Camera Name displays the camera that is selected from the Cameras combo box
11. Basic settings
12. Advanced settings
13. Get ROI button updates the current ROI values to X, Width, Y and Height
14. Full CCD button updates the full sensor size to X, Width, Y, and Height
15. X represents the X coordinate of an ROI
16. Width represents the width of an ROI
17. Y represents the Y coordinate of an ROI
18. Height represents the height of an ROI
19. Single Capture button allows to capture a single frame
20. Live start button starts live view mode
21. Exposure Time sets the exposure time
22. Exposure Resolution sets the exposure resolution
23. Binning combo box allows to update different binning mode
24. Save Image to TIFF button allows to save a TIFF image to this LabVIEW project's folder
25. Histogram toggle switch turns on/off histogram
26. Live stop button stops live view mode
27. Live indicator turns on if live view mode is on
28. Cameras combo box indicates cameras connected
29. Capturing indicator turns on if the camera is capturing image(s)
30. 8 bit histogram
31. Scrollbar to change the buffer index
32. Dropped frames to display how many frames are dropped
33. Buffer index to display which image is displayed on the camera window



UI layout

1. Ports and Speeds combo box allows to update supported ports and speeds
2. Gain combo box allows to choose different gain settings
3. Sensor clearing mode combo box allows to update supported clear mode
4. Clear Cycles combo box allows to update the clear cycle number
5. Trigger Out Behaviour combo box allows to update supported expose out mode.
6. Trigger In Behaviour combo box allows to update supported trigger mode
7. Post Processing allows to update supported post processing features
8. Value corresponds to the value set to the selected post processing feature
9. Color toggle switch turns on and off color mode for color camera



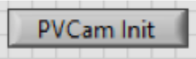
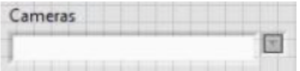
UI layout

1. Stream Saving mode
2. Stream Saving toggle switch to turn on and off Stream Saving mode
3. Stream Saving indicator to display if the switch is turned on or off
4. Format for Stream Saving mode. 0 for RAW and 1 for TIFF For color images only the TIFF format is supported
5. The control for the output path for Stream Saving images
6. The indicator for the output path for Stream Saving images

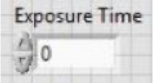
Steps to Run

Example 1: How to snap an image and save the image.

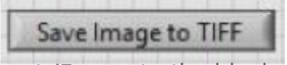
1. Press the run button to start

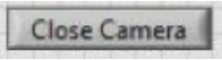
2. Press  PVCam Init to initialize the library followed by selecting a camera from  Cameras

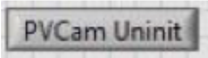
3. Press Open Camera to open the camera

4. Set the exposure value  Exposure Time 0

5. Press Single Capture to capture an image

6. Press  Save Image to TIFF to save a TIFF image. Makre sure the output path is set (Eg. go to the block diagram and select "Saving Image to Tiff" event set AcqBuffer::SavingPath. At default it is set as C:\Temps\ )

7. Press  Close Camera. The frame window is now closed

8. Press  PVCam Uninit to uninitialized the library

Example 2: How to initialize PVCAM, open camera, start Live, start histogram, stop live, save image, stop camera and uninitialize PVCAM.

Video: https://youtu.be/joB6MPc1PJg

Example 3: How to create and delete an ROI.

Video: https://youtu.be/qrirgamCaIY

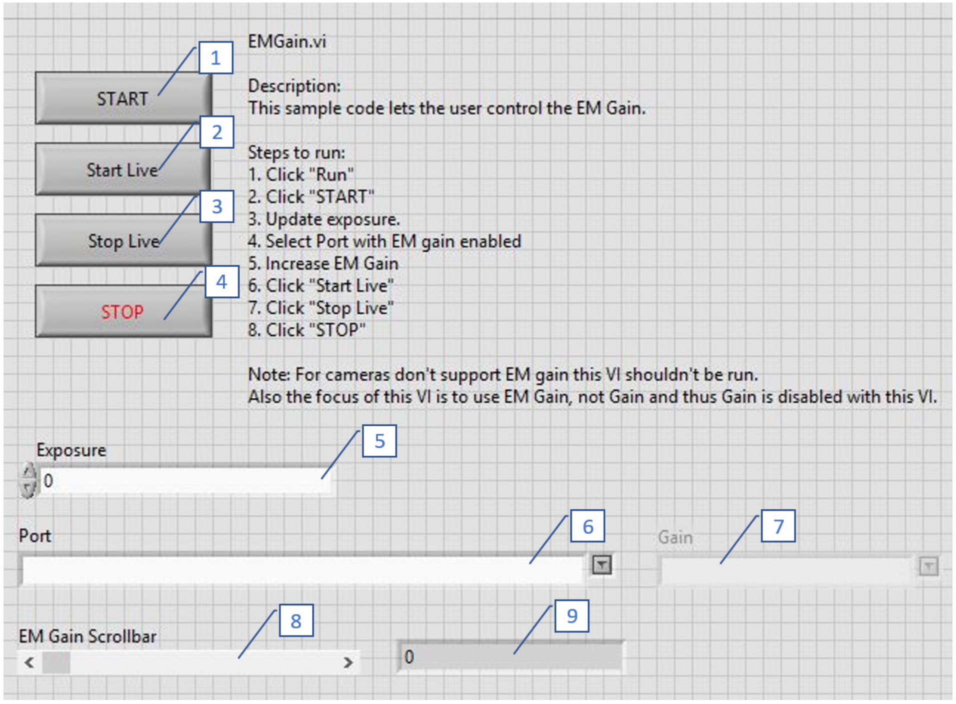
Example 4: How to select color mode, create an ROI and save image.

Video: https://youtu.be/WMJsX7yUUk0

Example 5: How to use stream saving mode.

Video: https://youtu.be/Ni5fx-dZCZk

EmGain.vi



EMGain.vi

Description:
This sample code lets the user control the EM Gain.

Steps to run:
1. Click "Run"
2. Click "START"
3. Update exposure.
4. Select Port with EM gain enabled
5. Increase EM Gain
6. Click "Start Live"
7. Click "Stop Live"
8. Click "STOP"

Note: For cameras don't support EM gain this VI shouldn't be run.
Also the focus of this VI is to use EM Gain, not Gain and thus Gain is disabled with this VI.

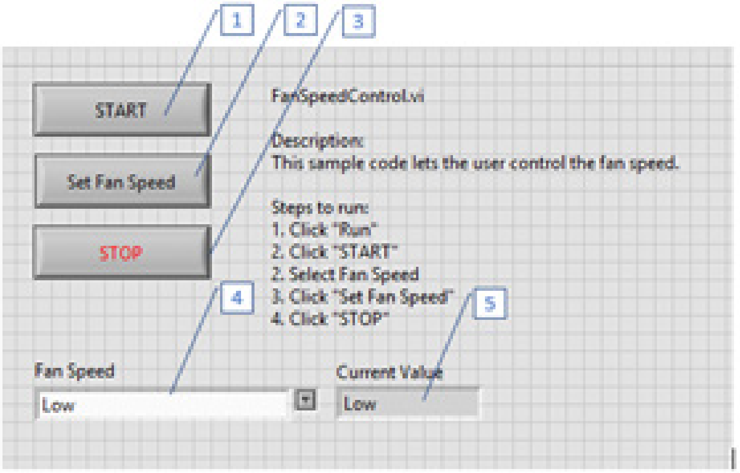
UI layout

1. START button starts the program
2. Start Live button starts live mode
3. Stop Live button stops live mode
4. STOP button ends the program
5 Exposure sets the exposure time
6. Port allows to update supported ports
7. Gain displays the supported gain
8. EM Gain Scrollbar allows to select supported gain
9. The indicator displays the gain value

How to run

Video: https://youtu.be/f0MfU014LL0

Fan SpeedControl.vi



FanSpeedControl.vi

Description:
This sample code lets the user control the fan speed.

Steps to run:
1. Click "Run"
2. Click "START"
2. Select Fan Speed
3. Click "Set Fan Speed"
4. Click "STOP"

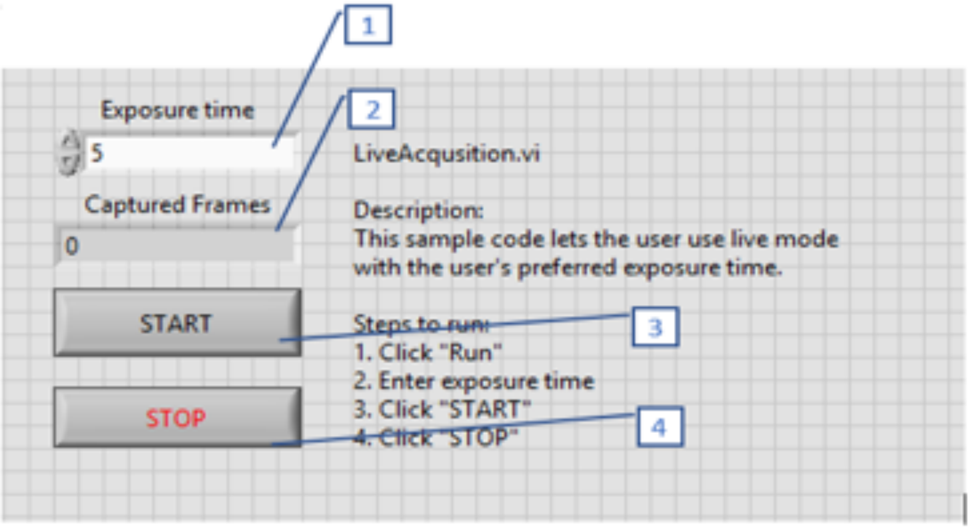
UI layout

1. START button starts the program
2. Set Fan Speed button sets the fan speed
3. STOP button to stop the program
4. Fan Speed combo box displays available fan speed
5. Current Value displays the current speed

How to run

Video: https://youtu.be/JPQXPqpj3L4

Live Acquisition.vi



Exposure time: 5

Captured Frames: 0

LiveAcqusition.vi

Description:
This sample code lets the user use live mode with the user's preferred exposure time.

Steps to run:
1. Click "Run"
2. Enter exposure time
3. Click "START"
4. Click "STOP"

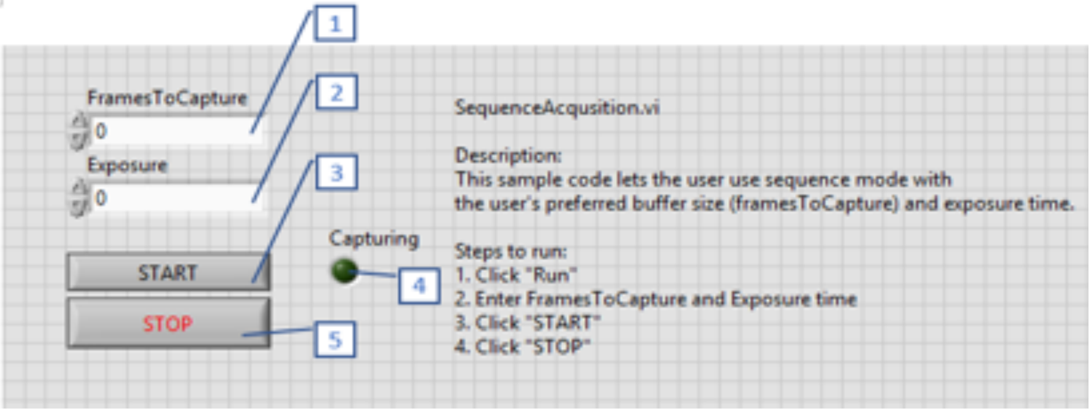
UI layout

1. Exposure Time sets exposure time
2. Captured Frames displays the number of frames captured
3. START button starts live mode
4. STOP button stops the VI

How to run

Video: https://youtu.be/tKqNMddupl0

Sequence Acquisition.vi



FramesToCapture: 0

Exposure: 0

SequenceAcqusition.vi

Description:
This sample code lets the user use sequence mode with the user's preferred buffer size (framesToCapture) and exposure time.

Steps to run:
1. Click "Run"
2. Enter FramesToCapture and Exposure time
3. Click "START"
4. Click "STOP"

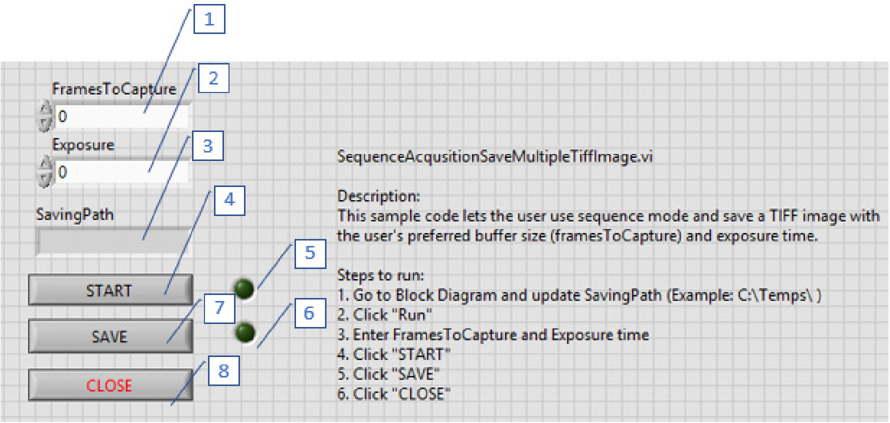
UI Layout

1. FramesToCapture sets X number of frames to be captured
2. Exposure Time sets exposure time
3. START button starts the program
4. Capturing indicator turns on while capturing
5. STOP button stops the program

How to run

Video: https://youtu.be/eaHH6zVI5pk

Sequence AcquisitionSave MultipleTiffImage.vi



SequenceAcqusitionSaveMultipleTiffImage.vi

Description:
This sample code lets the user use sequence mode and save a TIFF image with the user's preferred buffer size (framesToCapture) and exposure time.

Steps to run:
1. Go to Block Diagram and update SavingPath (Example: C:\Temps\ )
2. Click "Run"
3. Enter FramesToCapture and Exposure time
4. Click "START"
5. Click "SAVE"
6. Click "CLOSE"

UI Layout

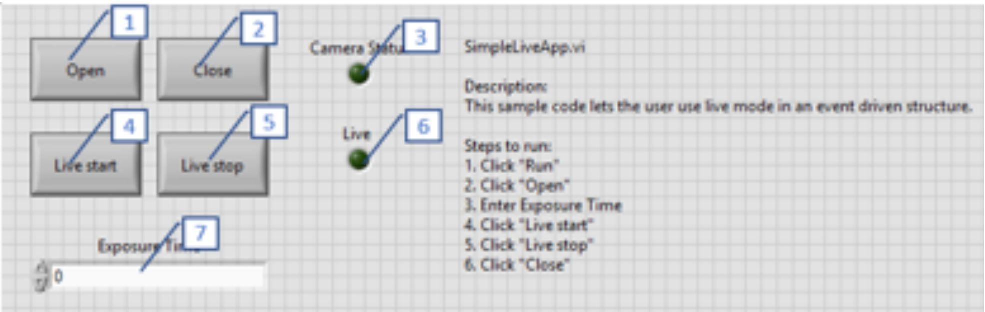
1. FramesToCapture sets X number of frames to be captured
2. Exposure Time sets exposure time
3. Saving path directory
4. START button starts the VI
5. Upon successful initialization the indicator turns on
6. After saving the indicator turns on
7. SAVE button saves multiple images

8, CLOSE button to end the VI

How to run

Video: https://youtu.be/DhA0nwUXP3M

SimpleLiveApp.vi



SimpleLiveApp.vi

Description:
This sample code lets the user use live mode in an event driven structure.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Click "Live start"
5. Click "Live stop"
6. Click "Close"

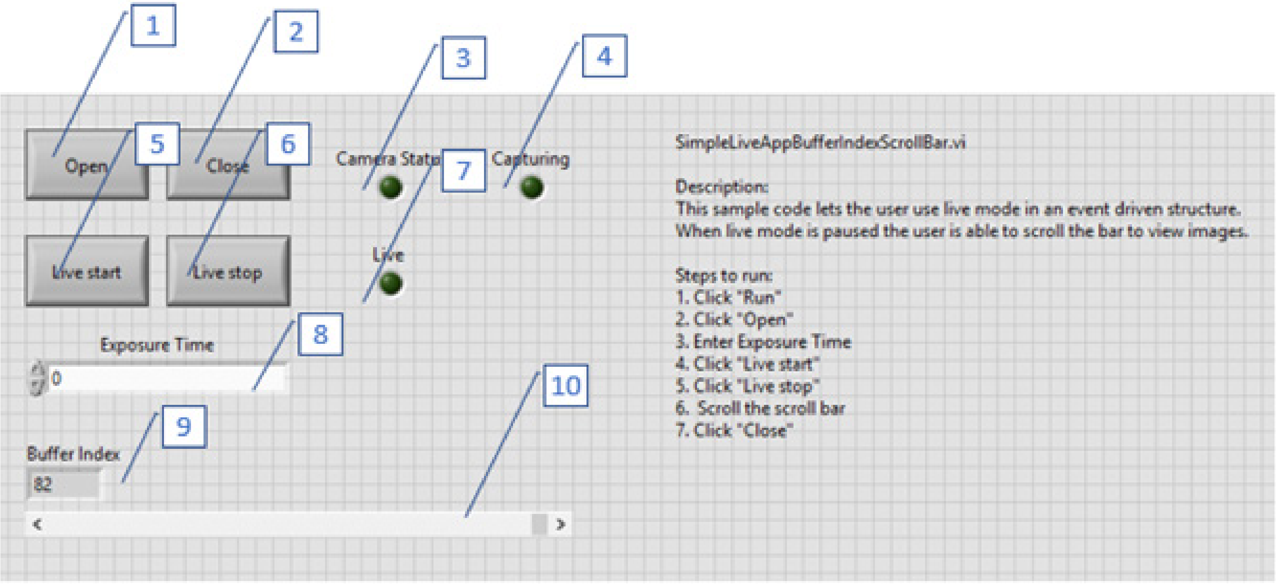
UI Layout

1. Open button will initialize PVCamNET and open the first index camera
2. Close button will close the camera and uninitialized PVCamNET
3. Camera Status turns on upon initialization
4. Live start button starts live mode
5. Live stop button stops live mode
6. Live indicator turns on during live mode
7. Exposure Time sets exposure time

How to run LiveAcqusition.vi

Video: https://youtu.be/roSviahC2Wk

SimpleLiveAppBufferIndexScrollBar.vi



SimpleLiveAppBufferIndexScrollBar.vi

Description:
This sample code lets the user use live mode in an event driven structure.
When live mode is paused the user is able to scroll the bar to view images.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Click "Live start"
5. Click "Live stop"
6. Scroll the scroll bar
7. Click "Close"

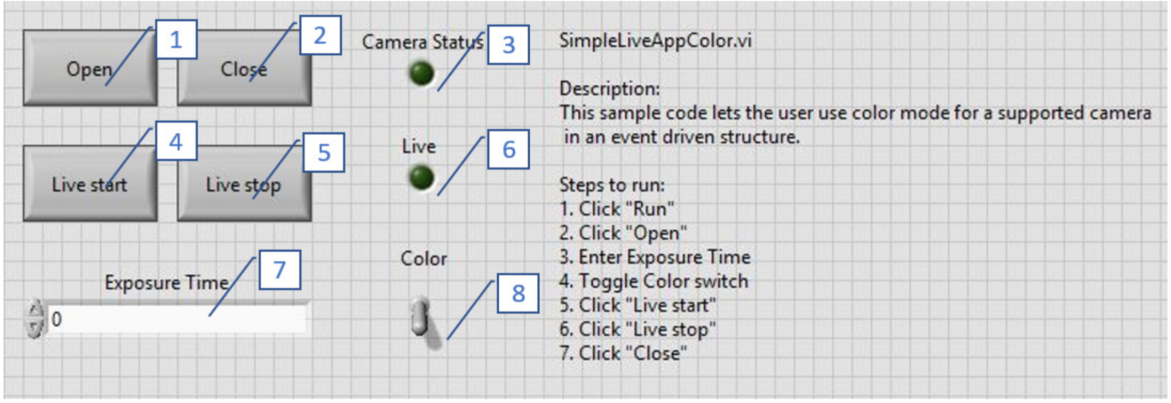
UI Layout

1. Open button initializes PVCamNET and open the first camera
2. Close button closes the camera and uninitializes PVCamNET
3. Camera Status indicator turns on after camera is successfully opened
4. Capturing indicator turns on while capturing images
5. Live start button starts live mode
6. Live stop button stops live mode
7. Live indicator turns on during live mode is on
8. Exposure Time sets exposure time
9. Buffer Index displays the current index in the circular buffer
10. Scollbar allows the user to change the buffer index when live mode is stopped

How to run

Video: https://youtu.be/rrMWdCo3skg

SimpleLiveAppColor.vi



SimpleLiveAppColor.vi

Description:
This sample code lets the user use color mode for a supported camera in an event driven structure.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Toggle Color switch
5. Click "Live start"
6. Click "Live stop"
7. Click "Close"

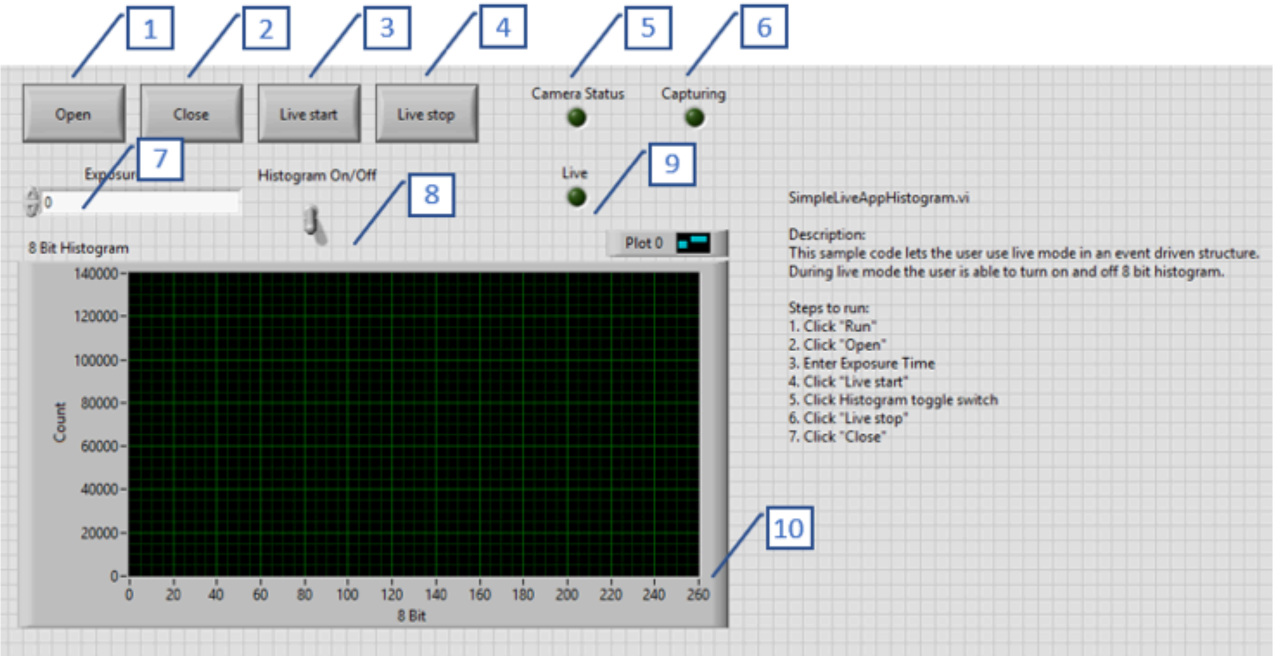
UI Layout

1. Open button initializes PVCamNET and open the first camera
2. Close button closes the camera and uninitializes PVCamNET
3. Camera Status indicator turns on after camera is successfully opened
4. Live start button starts live mode
5. Live stop button stops live mode
6. Live indicator turns on during live mode is on
7. Exposure Time sets exposure time
8. Color toggle switch allows to switch between mono and color

How to run

Video: https://youtu.be/ccVCpExAHBA

SimpleLiveAppHistogram.vi



SimpleLiveAppHistogram.vi

Description:
This sample code lets the user use live mode in an event driven structure.
During live mode the user is able to turn on and off 8 bit histogram.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Click "Live start"
5. Click Histogram toggle switch
6. Click "Live stop"
7. Click "Close"

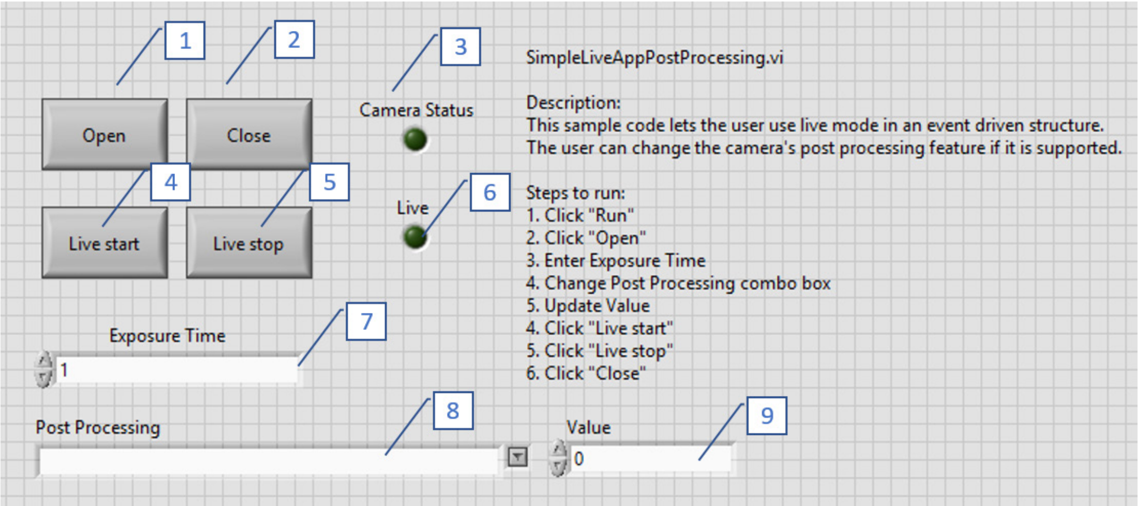
UI Layout

1. Open button initializes PVCamNET and open the first camera
2. Close button closes the camera and uninitializes PVCamNET
3. Live start button starts live mode
4. Live stop button stops live mode
5. Camera Status indicator turns on after the camera is successfully opened
6. Capturing indicator turns on while capturing images
7. Exposure Time sets exposure time
8. Histogram On/Off toggle switch turns on/off the histogram
9. Live indicator turns on when live mode is on.
10. Histogram

How to run

Video: https://youtu.be/xqgfb0hM92s

SimpleLiveAppPostProcessing.vi



SimpleLiveAppPostProcessing.vi

Description:
This sample code lets the user use live mode in an event driven structure.
The user can change the camera's post processing feature if it is supported.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Change Post Processing combo box
5. Update Value
4. Click "Live start"
5. Click "Live stop"
6. Click "Close"

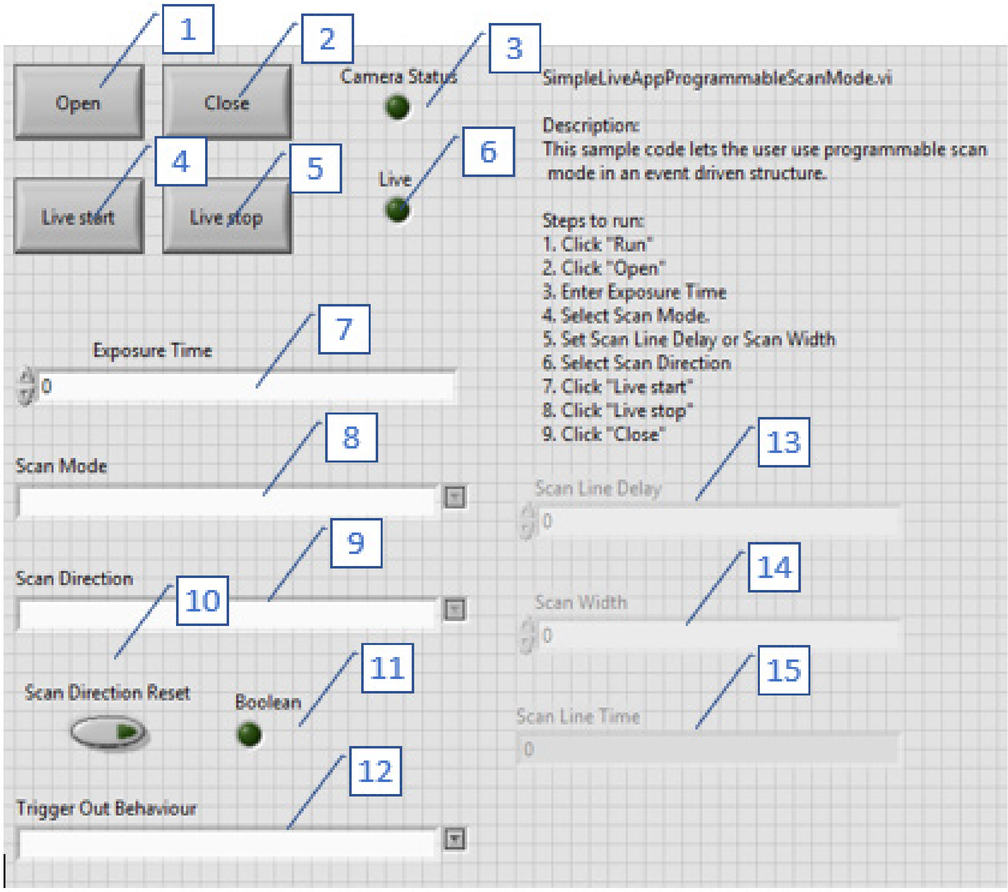
UI Layout

1. Open button initializes PVCamNET and open the first camera
2. Close button closes the camera and uninitializes PVCamNET
3. Camera Status indicator turns on after camera is successfully opened
4. Live start button starts live mode
5. Live stop button stops live mode
6. Live indicator turns on during live mode is on
7. Exposure Time sets exposure time
8. Post Processing combobox allows to select supported post processing features
9. Value sets the value for the selected post processing feature

How to run

Video: https://youtu.be/6G40wA9QcWI

SimpleLiveAppProgrammableScanMode.vi



SimpleLiveAppProgrammableScanMode.vi

Description:
This sample code lets the user use programmable scan mode in an event driven structure.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Select Scan Mode.
5. Set Scan Line Delay or Scan Width
6. Select Scan Direction
7. Click "Live start"
8. Click "Live stop"
9. Click "Close"

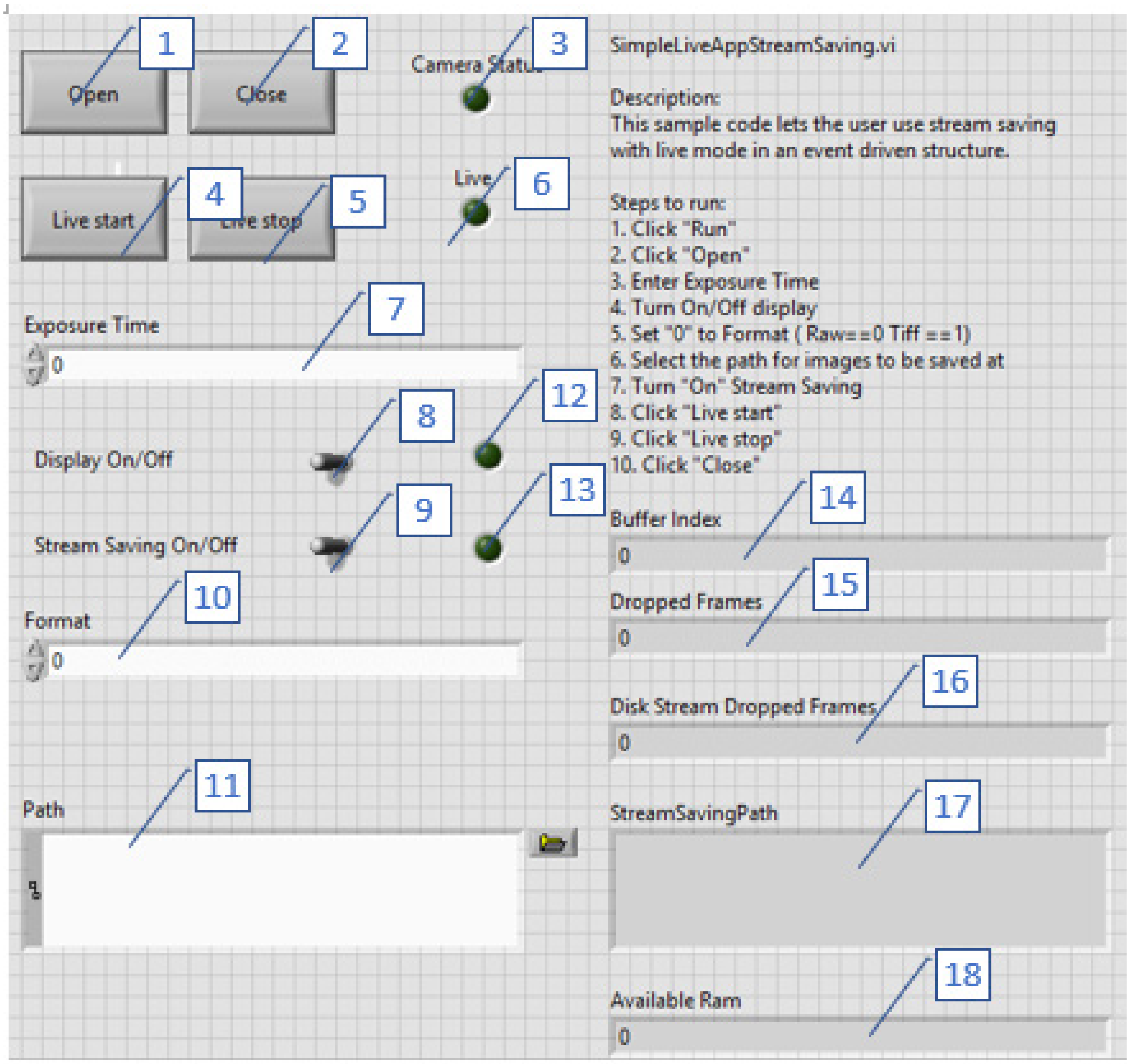
UI Layout

1. Open button initializes PVCamNET and open the first camera
2. Close button closes the camera and uninitializes PVCamNET
3. Camera Status indicator turns on after camera is successfully opened
4. Live start button starts live mode
5. Live stop button stops live mode
6. Live indicator turns on during live mode is on
7. Exposure Time sets exposure time
8. Programmable scan mode
9. Programmable scan direction
10. Programmable scan direction reset button
11. Programmable scan direction reset button indicator
12. Trigger out behaviour
13. Scan line delay configurable when scan line delay is chosen for the scan mode
14. Scan width configurable when scan width is chosen for the can mode
15. Scan line time (Read only)

How to run

Video: https://youtu.be/Yj4sRGaJXt4

SimpleLiveAppStreamSaving.vi



SimpleLiveAppStreamSaving.vi

Description:
This sample code lets the user use stream saving with live mode in an event driven structure.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Turn On/Off display
5. Set "0" to Format (Raw==0 Tiff ==1)
6. Select the path for images to be saved at
7. Turn "On" Stream Saving
8. Click "Live start"
9. Click "Live stop"
10. Click "Close"

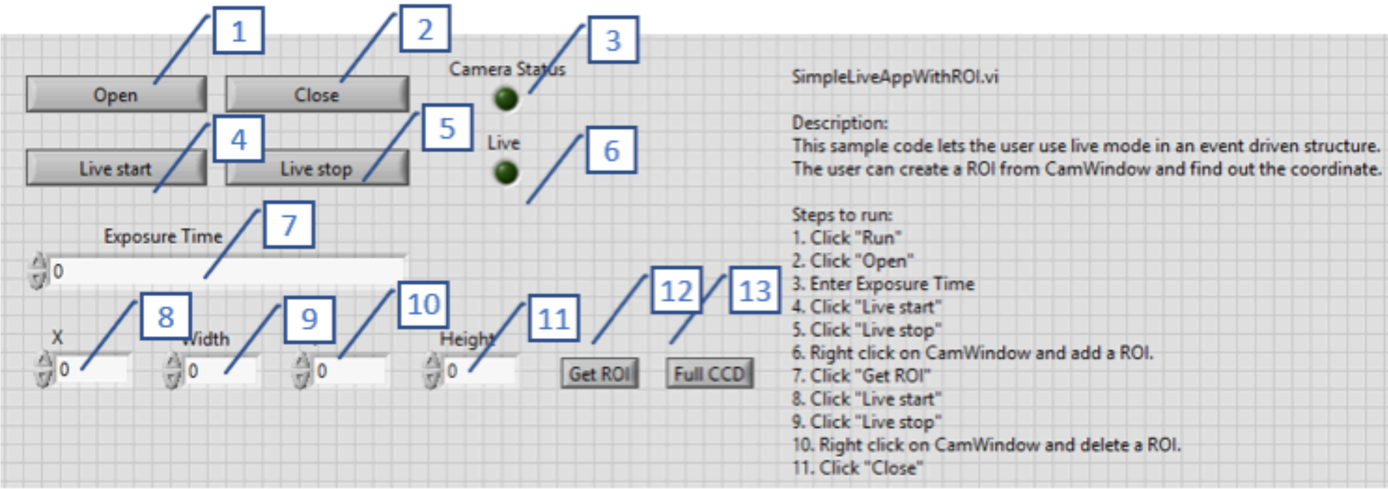
UI Layout

1. Open button initializes PVCamNET and open the first camera
2. Close button closes the camera and uninitializes PVCamNET
3. Camera Status indicator turns on after camera is successfully opened
4. Live start button starts live mode
5. Live stop button stops live mode
6. Live indicator turns on during live mode is on
7. Exposure Time sets exposure time
8. Display turn on/off toggle switch
9. Stream saving turn on/off toggle switch
10. Format (0 for RAW mono only and 1 for TIFF mono and cor)
11. The directory where stream saving images are saved
12. Display on/off indicator
13. Stream saving on/off indicator
14. Buffer index
15. Dropped frames
16. Stream saving dropped frames
17. Stream saving path
18. Available ram indicator

How to run

Video: https://youtu.be/4NkRfzgs4TQ

SimpleLiveAppWithROI.vi



SimpleLiveAppWithROI.vi

Description:
This sample code lets the user use live mode in an event driven structure.
The user can create a ROI from CamWindow and find out the coordinate.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Click "Live start"
5. Click "Live stop"
6. Right click on CamWindow and add a ROI.
7. Click "Get ROI"
8. Click "Live start"
9. Click "Live stop"
10. Right click on CamWindow and delete a ROI.
11. Click "Close"

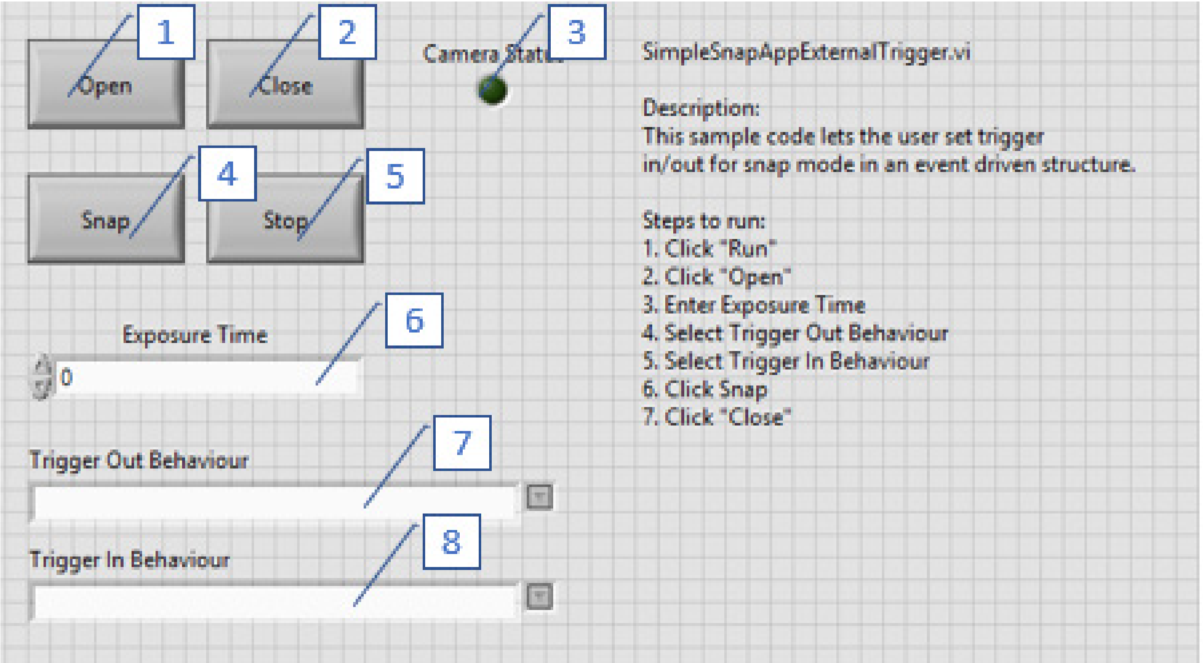
UI Layout

1. Open button initializes PVCamNET and open the first camera
2. Close button closes the camera and uninitializes PVCamNET
3. Camera Status indicator turns on after the camera is successfully opened
4. Live start button starts live mode
5. Live stop button stops live mode
6. Live indicator turns on while capturing images
7. Exposure Time sets exposure time
8. X sets the x coordinate of the ROI
9. Width sets the width of the ROI
10. Y sets the y coordinate of the ROI
11. Height sets the height of the ROI
12. Get ROI allows the user to get the current ROI and update X, Width, Y and Height
13. Full CCD allows the user to update the full CCD ROI and update X, Width, Y and Height

How to run

Video: https://youtu.be/EtCJDqJlkKs

SimpleSnapAppExternalTrigger.vi



SimpleSnapAppExternalTrigger.vi

Description:
This sample code lets the user set trigger in/out for snap mode in an event driven structure.

Steps to run:
1. Click "Run"
2. Click "Open"
3. Enter Exposure Time
4. Select Trigger Out Behaviour
5. Select Trigger In Behaviour
6. Click Snap
7. Click "Close"

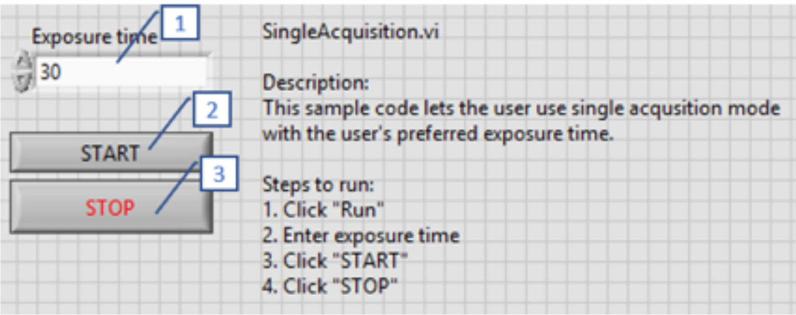
UI Layout

1. Open button initializes PVCamNET and open the first camera
2. Close button closes the camera and uninitializes PVCamNET
3. Camera Status indicator turns on after the camera is successfully opened
4. Snap start button starts snap mode
5. Live stop button stops live mode
6. Exposure Time sets exposure time
7. Trigger out behaviour
7. Trigger in behaviour

How to run

Video: https://youtu.be/VpVTkYvVYdM

SingleAcquisition.vi



30

SingleAcquisition.vi

Description:
This sample code lets the user use single acqusition mode with the user's preferred exposure time.

Steps to run:
1. Click "Run"
2. Enter exposure time
3. Click "START"
4. Click "STOP"

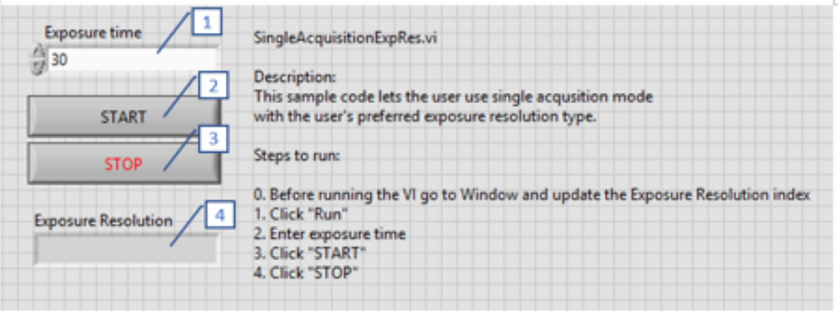
UI Layout

1. Exposure Time sets exposure time
2. START starts the VI
3. STOP button stops the VI

How to run

Video: https://youtu.be/uDhTSjs3JjE

SingleAcquisitionExpRes.vi



30

SingleAcquisitionExpRes.vi

Description:
This sample code lets the user use single acqusition mode with the user's preferred exposure resolution type.

Steps to run:

0. Before running the VI go to Window and update the Exposure Resolution index
1. Click "Run"
2. Enter exposure time
3. Click "START"
4. Click "STOP"

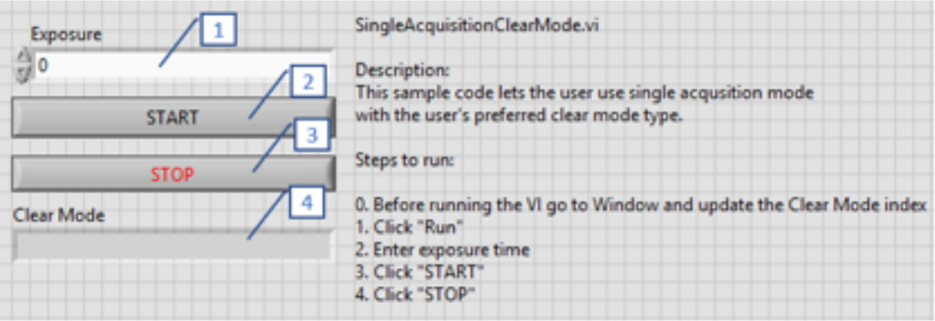
UI Layout

1. SingleAcquisitionExpRes
1. Exposure Time sets exposure time
2. START button starts the VI
3. STOP button stops the VI
4. Exposure Resolution indicator indicates the exposure resolution

How to run

Video: https://youtu.be/BoP7HMTxrHo

SingleAcquisitionClearMode.vi



SingleAcquisitionClearMode.vi

Description:
This sample code lets the user use single acqusition mode
with the user's preferred clear mode type.

Steps to run:

0. Before running the VI go to Window and update the Clear Mode index
1. Click "Run"
2. Enter exposure time
3. Click "START"
4. Click "STOP"

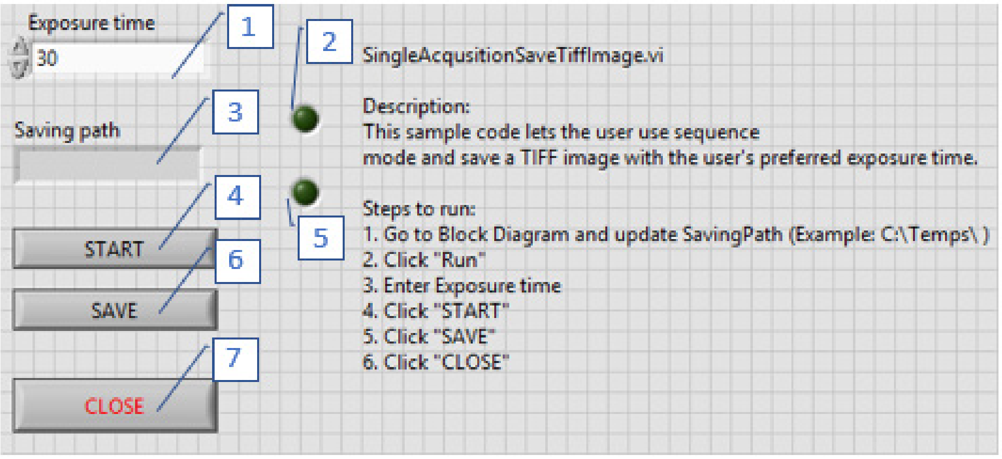
UI Layout

1. Exposure Time sets exposure time
2. START button starts the VI
3. STOP button stops the VI
4. Clear Mode indicates which clear mode is selected.

How to run

Video: https://youtu.be/xwGyDmR05WY

SingleAcquisitionSaveTiffImage.vi



SingleAcqusitionSaveTiffImage.vi

Description:
This sample code lets the user use sequence
mode and save a TIFF image with the user's preferred exposure time.

Steps to run:
1. Go to Block Diagram and update SavingPath (Example: C:\Temps\ )
2. Click "Run"
3. Enter Exposure time
4. Click "START"
5. Click "SAVE"
6. Click "CLOSE"

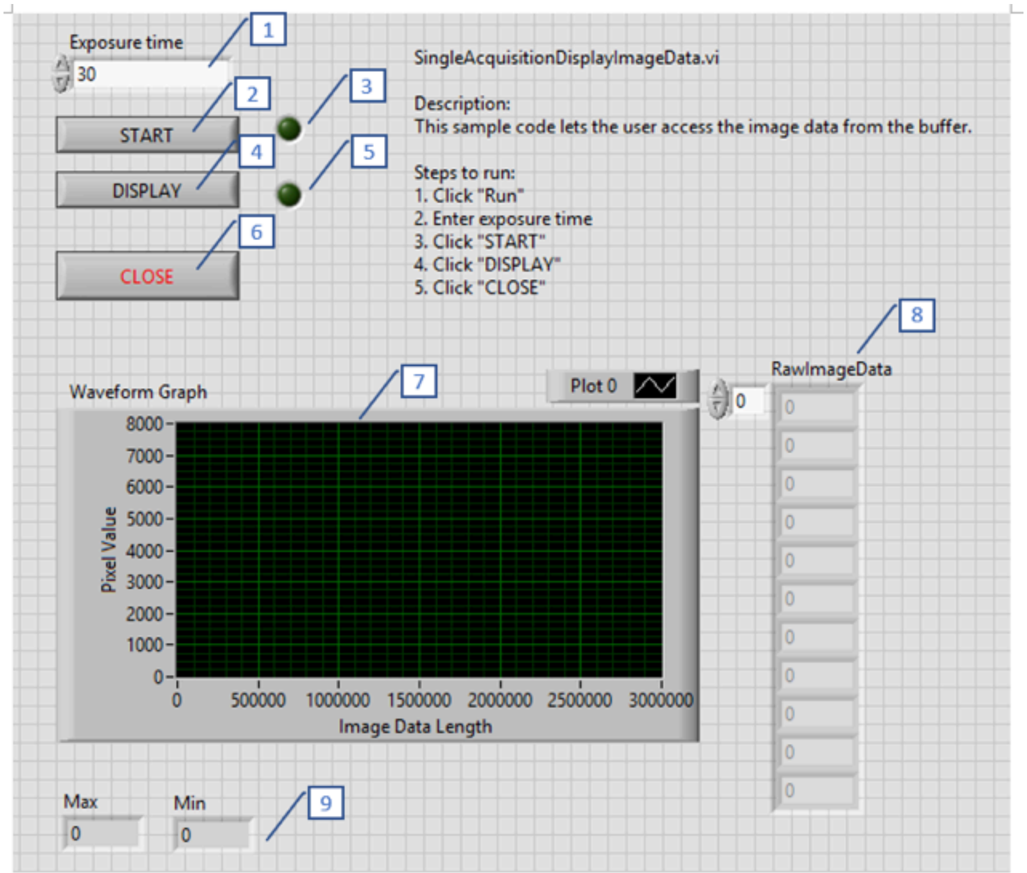
UI Layout

1. Exposure Time sets exposure time
2. The indicator turns on once the START button is pressed
3. Saving path directory
4. START button starts the VI
5. The indicator turns on once the image is saved.
6. SAVE button saves the image
7. CLOSE button stops the VI

How to run

Video: https://youtu.be/1t1SH9gqmP4

SingleAcquisitionDisplayImageData.vi



SingleAcquisitionDisplayImageData.vi

Description:
This sample code lets the user access the image data from the buffer.

Steps to run:
1. Click "Run"
2. Enter exposure time
3. Click "START"
4. Click "DISPLAY"
5. Click "CLOSE"

UI Layout

1. Exposure Time sets exposure time
2. START button starts the VI
3. The indicator turns on after the START button is pressed
4. DISPLAY button updates Waveform Graph, RawImageData, Max and Min indicators
5. The indicator turns on after the DISPLAY button is pressed
6. CLOSE button stops the VI
7. Waveform Graph displays the pixel value of the image
8. RawImageData displays the first 11 image pixel values
9. Max and Min display the highest and lowest pixel value of the image

How to run

Video: https://youtu.be/4CExXdG1HxM

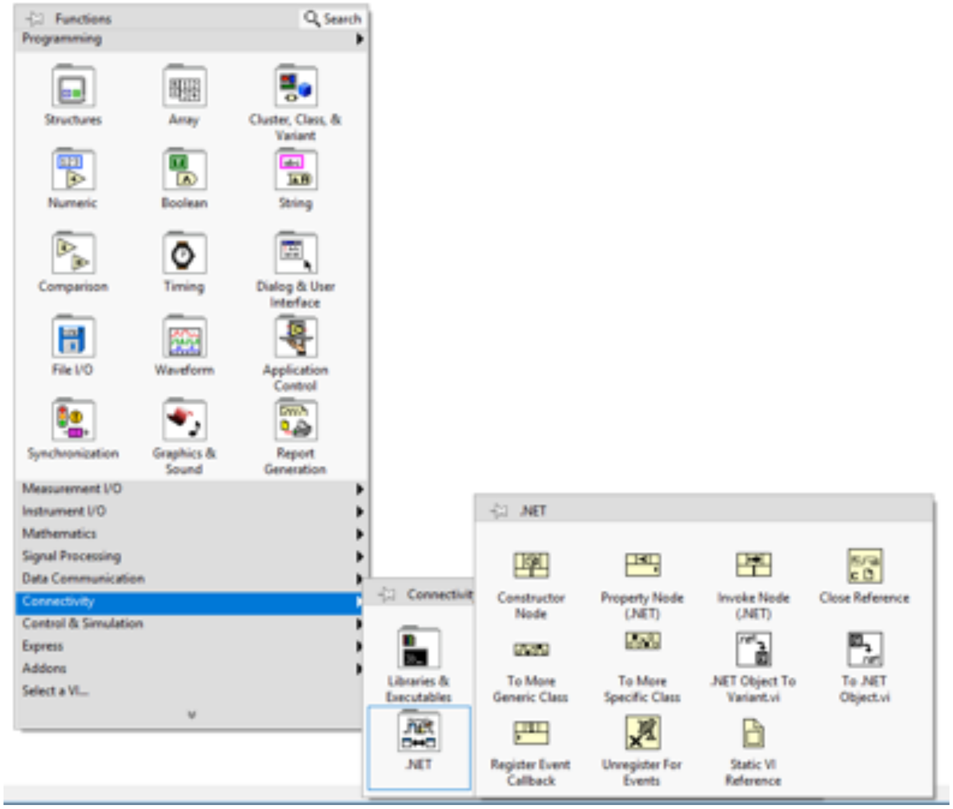
Chapter 4.

GETTING STARTED

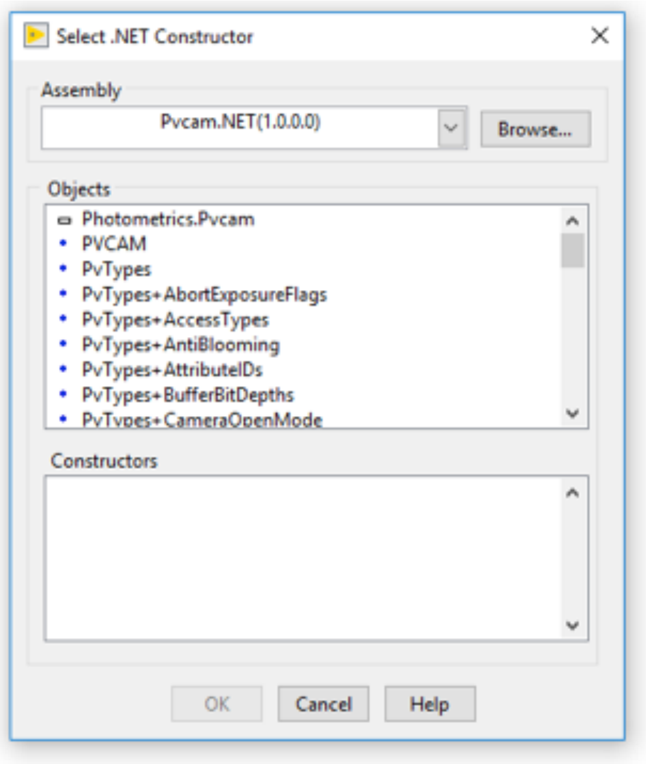
How to use the interface

If you are not familiar with the PVCamNET Assembly please go over its documentation and the example LabVIEW code such as “SingleAcquistion.vi” to understand the flow of function calls. Once you understand “SingleAcqusition.vi”, “SequenceAcqusition.vi”, “LiveAcquisition.vi”, please look at “SimpleLiveApp.vi” and “LiveAppWithROI.vi”. These two VIs are event driven apps, which will help you understand the structure of the demo app, “PVCamApp.vi”.

To access the PVCamNET classes, properties and functions, look under “Connectivity” in the User Libraries menu and click “.NET”. There is “Constructor Node“ and “Property Node“ and “Invoke Node” in the menu.



Once a node is clicked, a pop-up message will appear to select the .NET assembly. Browse and select PVCamNET. Select an object from the following list to access a property, a function or an object of the PVCamNET.



Assembly
Pvcam.NET(1.0.0.0)

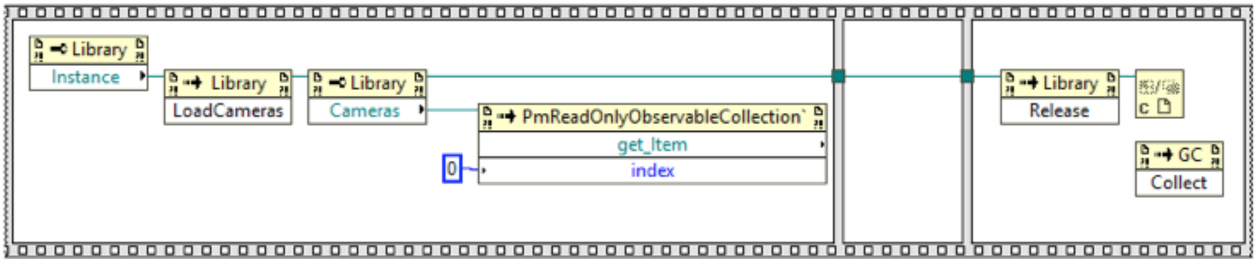
Objects
- Photometrics.Pvcam
- PVCAM
- PvTypes
- PvTypes+AbortExposureFlags
- PvTypes+AccessTypes
- PvTypes+AntiBlooming
- PvTypes+AttributeIDs
- PvTypes+BufferBitDepths
- PvTypes+CameraOpenMode

Constructors

Quick Start Guide

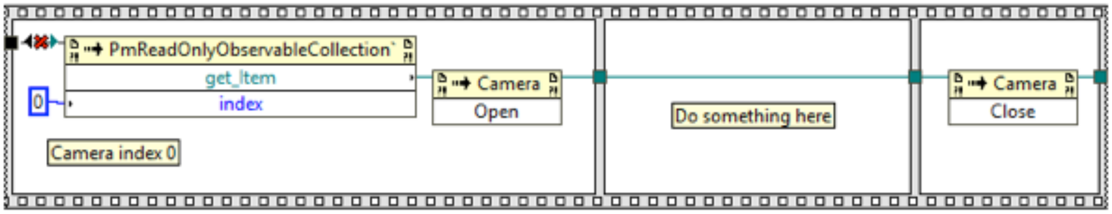
Open and close library

Before opening the camera, the library must be initialized by calling Library Instance. From Library Cameras the user can select which camera to open. Once the library is finished then the user must call Library Release, remove its reference and call GC Collect. (Example: SingleAcqusition.vi)



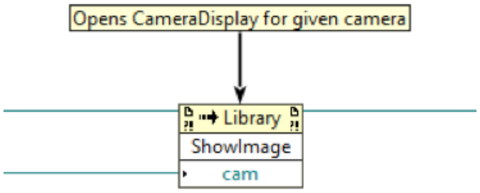
Open and close camera

After initializing the library one of the cameras connected to the PC can be opened via Camera Open. When the camera is finished then Camera Close must be called to close the camera. (Example: SingleAcqusition.vi)



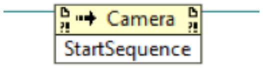
Display image

After initializing the library, the PVCamNET library can generate a window to display acquired images. This is done by accessing Library Show Image connecting the Library instance and the camera instance. (Example: SingleAcqusition.vi)



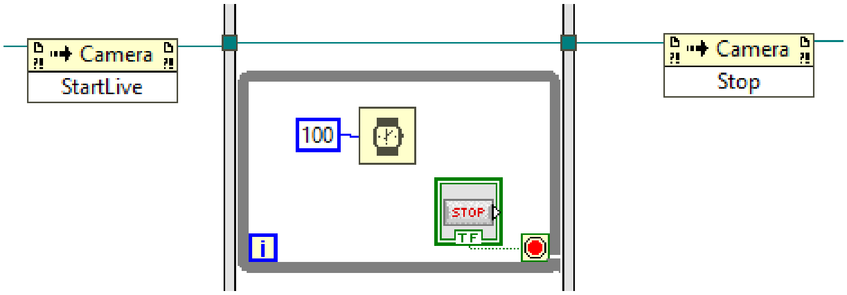
Snap

After opening the camera, Camera StartSequence must be called to snap an image. (Example: SingleAcqusition.vi)



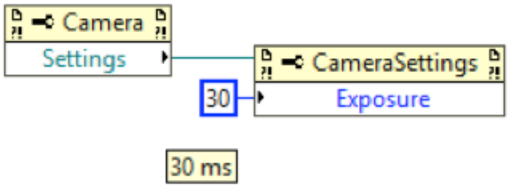
Live start and stop

After opening the camera, Camera StartLive must be called to start a live acquisition. Once finished, Camera Stop must be called to stop live acquisitions. (Example: LiveAcqusition.vi)



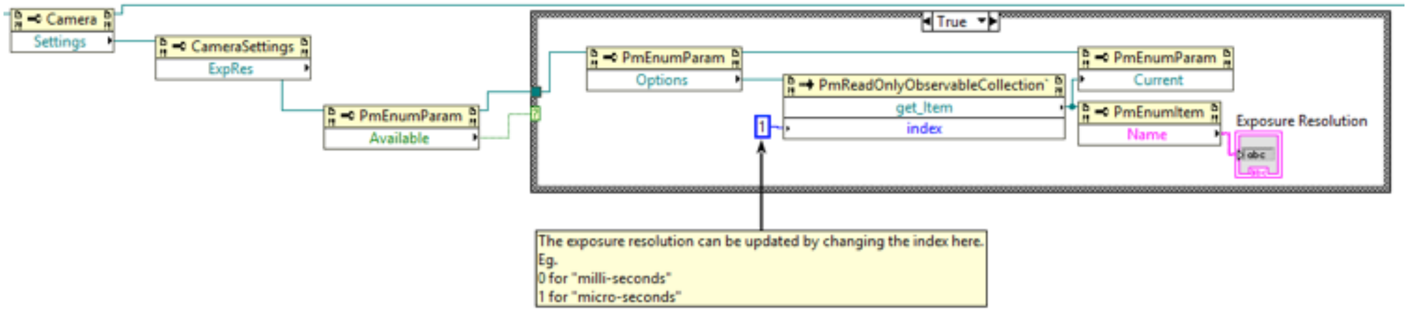
Set up exposure time

After opening the camera the user can update the camera's exposure time by accessing Camera Settings. (Example: SingleAcqusition.vi)



Set up exposure resolution

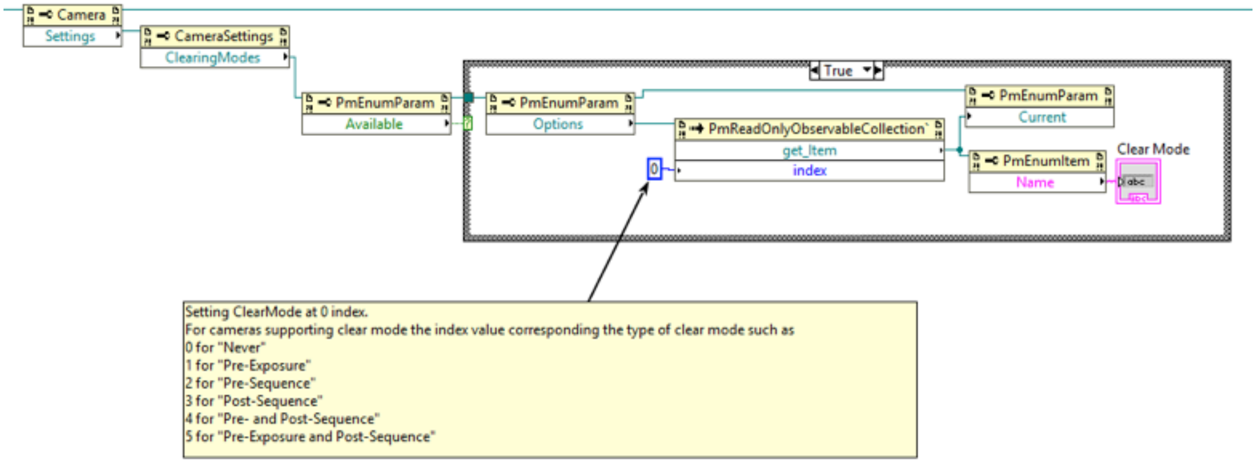
After opening the camera, access Camera Settings ExpRes. Check if the camera supports Exposure Resolution with Camera Settings ExpRes followed by PmEnumParam Available. If the camera is supported, then access PmEnumParamOptions followed by PmReadOnlyObservableCollection. Update the index value. (Example: SingleAcqusitionExpRes.vi)



The exposure resolution can be updated by changing the index here.
Eg.
0 for "milli-seconds"
1 for "micro-seconds"

Set up clear mode

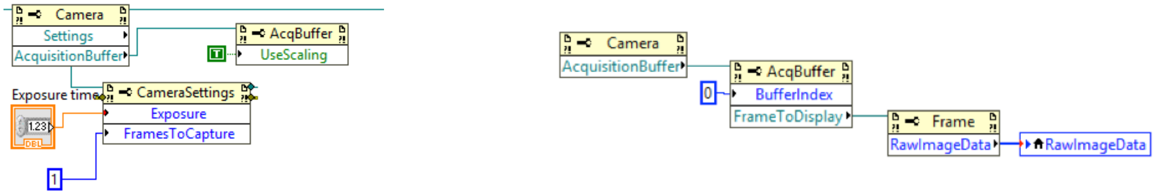
After opening the camera, access Camera Settings ClearingModes. Check if the camera supports Clearing Modes with Camera Settings ClearingModes followed by PmEnumParam Available. If the camera is supported, then access PmEnumParamOptions followed by PmReadOnlyObservableCollection. Update the index value. Note that some cameras that support clearing mode don't necessarily have the same index number for the clear mode types as the example below (Example: SingleAcqusitionClearMode.vi)



Setting ClearMode at 0 index.
For cameras supporting clear mode the index value corresponding the type of clear mode such as
0 for "Never"
1 for "Pre-Exposure"
2 for "Pre-Sequence"
3 for "Post-Sequence"
4 for "Pre- and Post-Sequence"
5 for "Pre-Exposure and Post-Sequence"

Accessing images

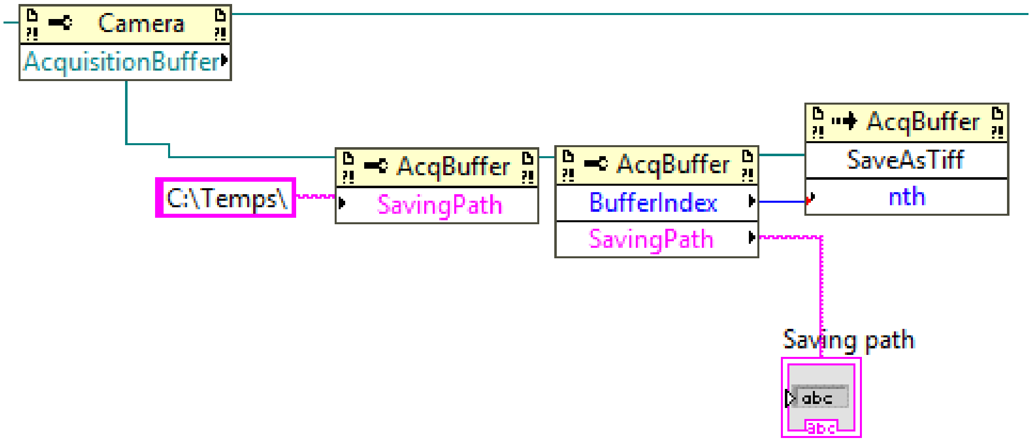
From the camera window the user can view acquired images, but to access the image data from LabVIEW the user must: 1) enable UseScaling from AcqBuffer and 2) access Camera AcqusitionBuffer, select BufferIndex, and Frame's RawImageData from FrameToDisplay. (Example: SingleAcquisitionDisplayImageData.vi)



Saving images

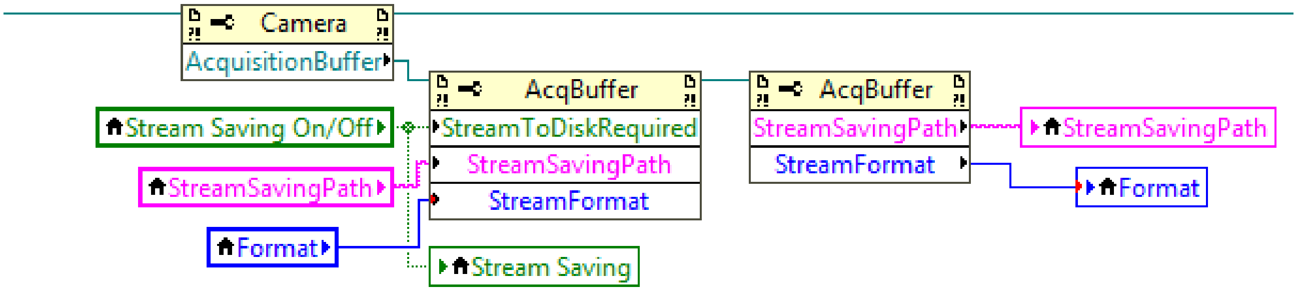
There are two ways to save the image data. One is to save image(s) given image buffer index and the other is to use stream saving.

For the first method the user must access Camera AcqusitionBuffer, select BufferIndex, and finally access Frame's SaveAsTiff from FrameToDisplay. The user also has to set the output directory from AcqBuffer SavingPath. (Example: SingleAcquisitionSaveTiffImage.vi SequenceAcquisitionSaveMultipleTiffImage.vi)



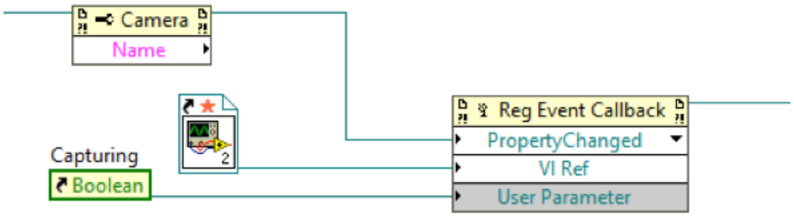
For stream saving mode the user must set the following properties in AcquisitionBuffer:
- StreamToDiskRequired to be true
- StreamSavingPath to be where the user wants output images to be. The folder must exist.
- StreamFormat to be either 0 or 1.
  - 0 for RAW format. Mono is supported.
  - 1 for TIFF format. Mono and color are both supported.

(Example: SimpleLiveAppStreamSaving.vi)

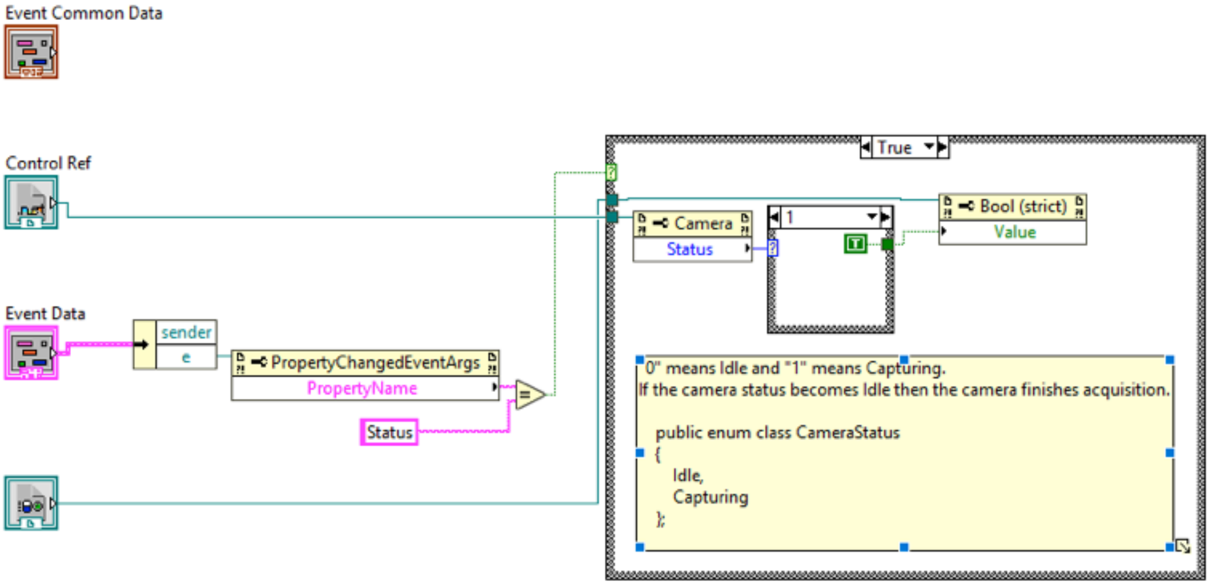


Camera state

The PVCamNET library can tell the camera's state; either Idle or Capturing. The state can be found from accessing Camera. Access Reg Event Callback and connect from Camera to PropertyChanged and create a callback VI (Example: SimpleLiveAppHistogram.vi)



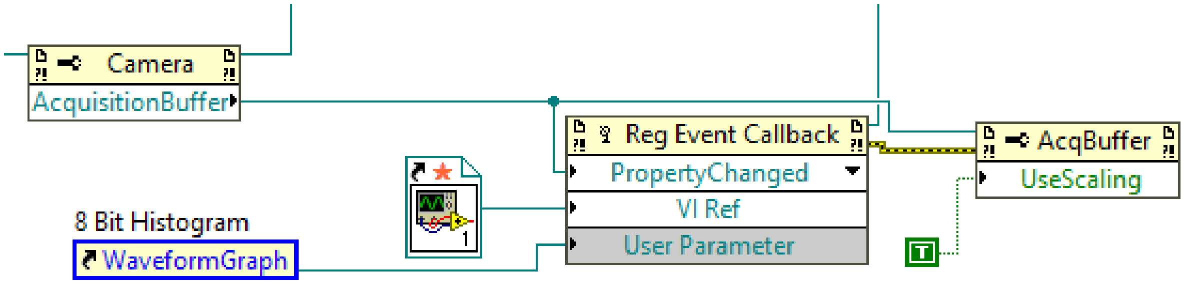
In the callback VI, connect EventData to sender and to PropertyChangeEventArgs. Create a string "Status" and create a case if the PropertyName and the string are equal. Then update the Boolean value with the Camera Status. After using the callback make sure to deregister the callback and call GC Collect on the VI that is used to register the callback to free the memory used by the callback VI.
(Example: PropertyChangedEventCallbackForCameraState.vi)



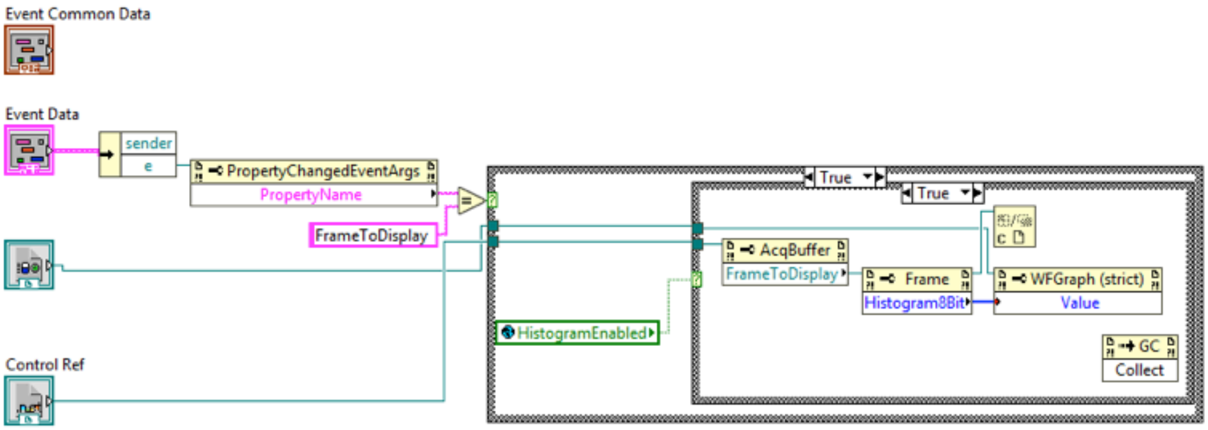
```
0" means Idle and "1" means Capturing.
If the camera status becomes Idle then the camera finishes acquisition.

  public enum class CameraStatus
  {
      Idle,
      Capturing
  };
```

Histogram

To display a histogram of image data the user must enable UseScaling and register a callback event so that when every new frame arrives the callback event will display the new frame's histogram. To register a callback, access AcqusitionBuffer, select PropertyChanged, create a callback VI, and pass the reference WaveformGraph to the User Parameter (Example: SimpleLiveAppHistogram.vi)

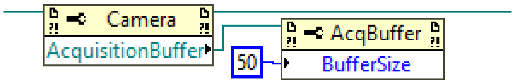


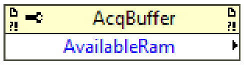
Inside the callback VI, connect PropertyChangedEventArgs select PropertyName. Create a string "FrameToDisplay". If the PropertyName's output is equal to the string, and if HistogramEnabled is turned on, the Frame Histogram8Bit can be accessed from AcqBuffer FrameToDisplay to display on the wave form graph. After using the callback make sure to deregister the callback and call GC Collect on the VI that is used to register the callback to free the memory used by the callback VI. (Example: PropertyChangedEventCallbackForHistogram.vi)



BufferSize

The user is able to configure the PVCamNET's secondary buffer size from AcqusitionBuffer and access BufferSize.





AvailableRam

This property can be used to check how much RAM is available for the processing buffer. The processing buffer size (1 frame size in bytes = width*height*2 => 2 for 16 bits). The processing buffer size must be set in such way that acquisition will run long enough.

Examples

How to open the project

Video: https://youtu.be/HVkPZ94fkiQ

- demonstrate how to open the project after a successful installation

How to create single capture VI

The following videos will demonstrate how to create single capture example:

Part1

Video: https://youtu.be/gCYHL9tmmG4

- access the PVCamNET library from a VI
- initialize the library, load cameras, select a camera, open the camera, shows
- the window set camera settings

Part 2

Video: https://youtu.be/yWWCZhmCURU

- start sequence, close the camera, release the library
- group nodes with a flat sequence structure

Part 3

Video: https://youtu.be/Y4eDRP_N008

- run the VI

How to create live capture VI with callback

The following videos will demonstrate how to create live capture example:

Part1

Video: https://youtu.be/L0nNCRuZsjo

- access the PVCamNET library from a VI
- initialize the library, load cameras, select a camera, open the camera, shows the window set camera settings, start live, stop live, close the camera,

Part 2

Video: https://youtu.be/cvn7bstmQVU

- group nodes together with a flat sequence structure
- add a while loop between start and stop live

Part 3

Video: https://youtu.be/oQg11icr1Dk

- test the VI before adding the callback

Part 4

Video: https://youtu.be/p7-nJ8KE1kw

- add the callback vi to display captured frames

Part 5

Video: https://youtu.be/TnRGWEWqmg0

- run the VI

Chapter 5.

FEEDBACK

Photometrics welcomes feedback at any time.
https://www.photometrics.com/support/contact-support

If you have problems, please gather the following infomation and report to the customer service. Before you report, please check your system requirements, known issues and limitations.

- PVCamNET.dll version (Right click on the dll and look for File/Product version from Properties [eg. 1.0.126.0])
- LabVIEW version (Go to Help -> About [eg. Version 18.0f2 (16-bit)])
- Windows version (Go to System -> Windows edition [eg. Windows 10 Pro 64 bit])
- PC configuration (CPU and RAM size)

PHOTOMETRICS®
Scientific CMOS, EMCCD and CCD Cameras

www.photometrics.com
Main Phone: +1 520.889.9933
Support: +1 604.530.5800 / +1 800.874.9789