pytelicam Samples

(Python) Manual

Version 1.0.0 (2024/12/13)

Toshiba Teli Corporation

Information contained in this document is subject to change without prior notice.

DAA05081A

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

Contents

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

1. Introduction

This document describes how to use pytelicam sample codes. Sample codes can be used by installing TeliCamSDK. TeliCamSDK can be obtained from TeliCamSDK home page.

The knowledge about GenICam, GigE Vision, USB3 Vision, and IIDC2 register map will be useful to understand sample codes. The detail or latest information of these standards can be obtained from home page of these standards.

GenICam	https://www.emva.org/standards-technology/genicam/
GigE Vision	https://www.automate.org/a3-content/vision-standards-gige-vision
USB3 Vision	https://www.automate.org/a3-content/usb3-vision-standard
IIDC2	http://jiia.org/mv_dl/iidc2/

2. File path of sample codes

These sample codes are located in the “Samples” folder of pytelicam.

Refer to the following “pytelicam User Guide Eng.pdf” for the installation method of TeliCamSDK.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3. Sample Codes of Python

pytelicam provides some sample codes projects in the following table for Python user's reference. More sample sources will be provided in our home page subsequently.

Sample name	UI	Function
get_camera_information	CUI	Display of camera information.
configure_camera_feature		Acquisition and setting of parameters using the functions of CameraControl class.
configure_camera_feature_using_genicam		Acquisition and setting of parameters using the functions of GenApiWrapper class.
grab_image_callback		Continuous capture of images using the Callback function.
grab_image_callback_using_trigger		Image capture of software trigger using the Callback function.
grab_image_callback_opencv	GUI	Continuous capture of images using the Callback function and display images using the OpenCV.
grab_next_image	CUI	Continuous capture of images using the get_next_image function.
grab_next_image_using_trigger		Image capture of software trigger using the get_next_image function.
grab_next_image_opencv	GUI	Continuous capture of images using the get_next_image function and display images using the OpenCV.
grab_image_opencv-like		Continuous capture of images using the get_next_image function and display images using the OpenCV. It uses a sample class that has function like the VideoCapture of the OpenCV.
grab_current_image	CUI	Continuous capture of images using the get_current_buffered_image function.
grab_current_image_using_trigger		Image capture of software trigger using the get_current_buffered_image function.
Grab_current_image_opencv	GUI	Continuous capture of images using the get_current_buffered_image function and display images using the OpenCV.
grab_buffered_image	CUI	Continuous capture of images using the get_current_buffer_index function and the get_buffered_image function.
grab_buffered_image_using_trigger		Image capture of software trigger using the get_current_buffer_index function and the get_buffered_image function.
Grab_buffered_image_opencv	GUI	Continuous capture of images using the get_current_buffer_index function and the get_buffered_image function and display images using the OpenCV.
grab_camera_event	CUI	Get camera event.
grab_chunk_data		Acquisition the chunk.
get_camera_feature_list		Display a list of camera features.
multi_camera_opencv	GUI	Display images of multi cameras using the OpenCV.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.1. get_camera_information

This sample code describes how to display information of all connected cameras.

```
<System information>
  TeliCamApi version  : 3.0.4.1
6 camera(s) found.
<Camera0 information>
  Cam type              : CameraType.U3v
  Cam vendor            : Toshiba-Teli
  Cam model             : BU040MC
  Cam serial number     : ES #2
  Cam version           : 4.1.0.1
  Cam user defined name : ab
  Cam display name      : BU Series
  TL vendor             : Toshiba Teli Corporation
  TL model              : TeliU3vCamApi
  TL version            : 2.0.7.1
  TL display name       : Toshiba Teli Standard USB3 Vision Interface
  TL-IF display name    : USB3 Vision Interface
```

[Main functions of this sample code]

get_camera_information()

[Remarks]

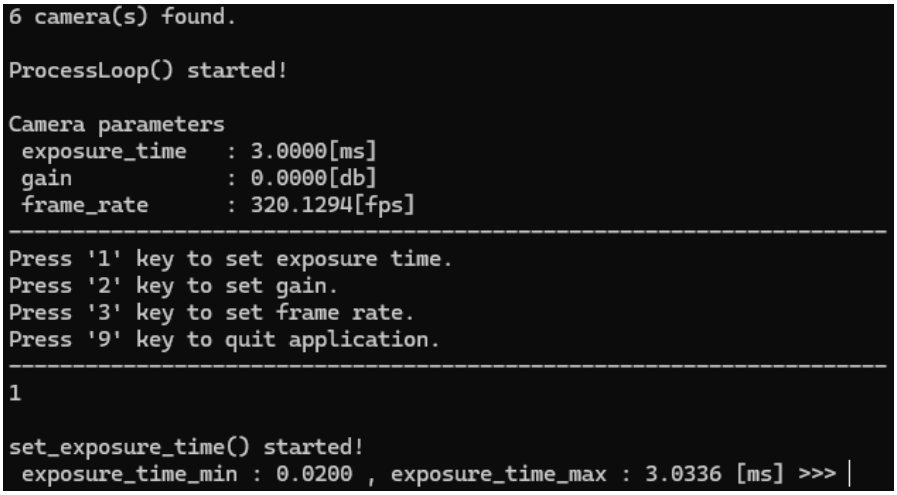
The get_camera_information function gets camera information.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.2. configure_camera_feature

This sample code describes how to acquire and set camera parameters using the functions of CameraControl class.



[Main functions of this sample code]

get_exposure_time_min_max()、get_exposure_time()、set_exposure_time()、get_gain_min_max()、get_gain()、set_gain()、get_acquisition_frame_rate_min_max()、get_acquisition_frame_rate()、set_acquisition_frame_rate()

[Remarks]

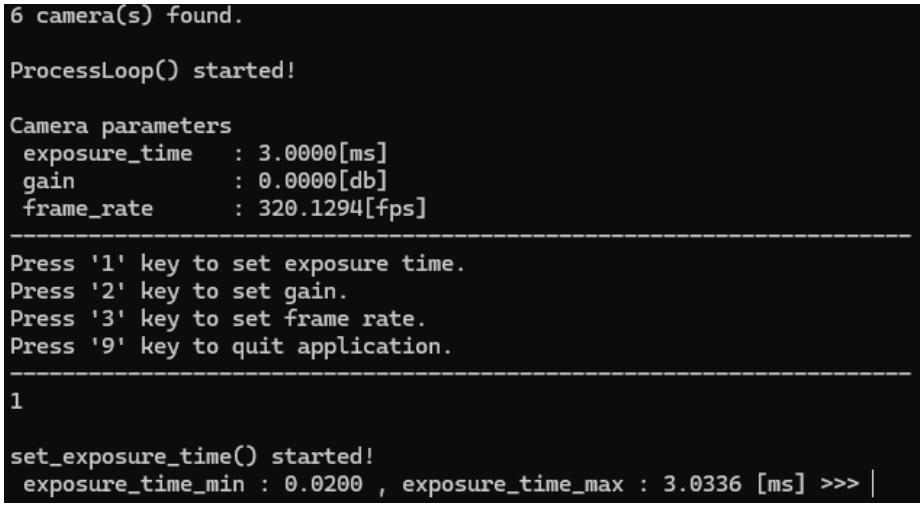
The functions of CameraControl class can process faster than when using the functions of GenApiWrapper class. However, the functions of CameraControl class are not available for all camera features.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.3. configure_camera_feature_using_genicam

This sample code describes how to acquire and set camera parameters using the functions of GenApiWrapper class.



[Main functions of this sample code]

get_float_min()、get_float_max()、set_float_value()

[Remarks]

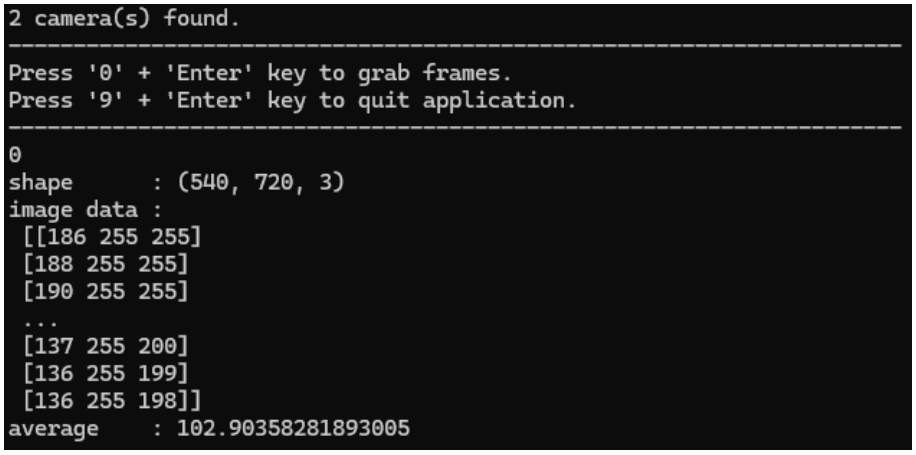
The functions of GenApiWrapper class can be used to camera features that are not prepared by the functions of CameraControl class.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.4. grab_image_callback

This sample code describes how to continuously capture images using the callback feature.



[Main functions of this sample code]

set_callback_image_acquired()、set_callback_image_error()、set_callback_buffer_busy()

[Remarks]

The set_callback_image_acquired function can be used to register a callback feature. The registered callback feature will be executed when new image data is stored in the stream ring buffer inside this API.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
- Using “get_next_image()”.
  Refer to the grab_next_image sample code.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.
  Refer to the grab_buffered_image sample code.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

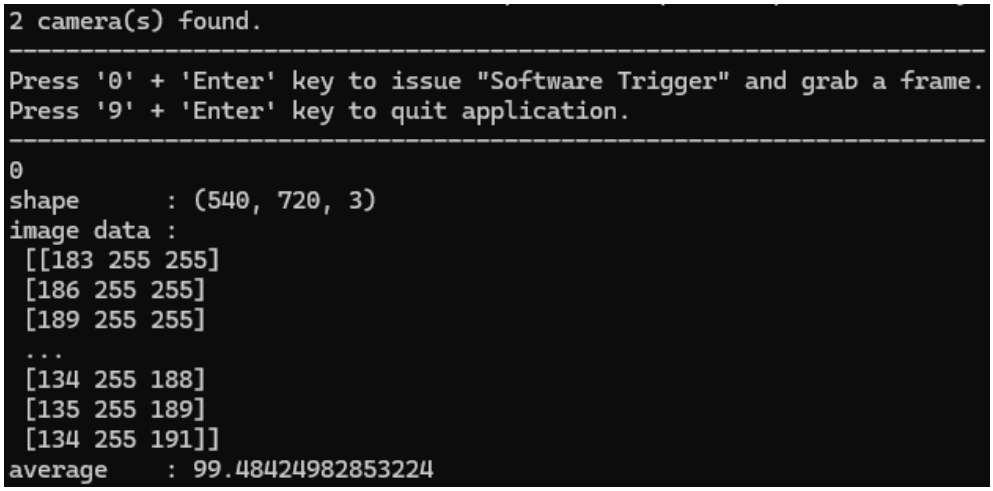
If capture images are got to continuously, the “set_trigger_mode” function is set to false, allowing to acquire images from the camera without waiting for a trigger.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.5. grab_image_callback_using_trigger

This sample code describes how to capture an image using the Callback function.



[Main functions of this sample code]

set_callback_image_acquired()、set_callback_image_error()、set_callback_buffer_busy()

[Remarks]

The set_callback_image_acquired function can be used to register a callback feature. The registered callback feature will be executed when new image data is stored in the stream ring buffer inside this API.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
- Using “get_next_image()”.
  Refer to the grab_next_image_using_trigger sample code.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image_using_trigger sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.
  Refer to the grab_buffered_image_using_trigger sample code.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

The “set_trigger_mode” function can set both the software trigger and the hardware trigger.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.6. grab_image_callback_opencv

This sample code describes how to continuously capture images using the Callback function and display images using the OpenCV. This image is test pattern (ColorBar).

[Main functions of this sample code]

set_callback_image_acquired()、set_callback_image_error()、set_callback_buffer_busy()

[Remarks]

The set_callback_image_acquired function can be used to register a callback feature. The registered callback feature will be executed when new image data is stored in the stream ring buffer inside this API.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
- Using “get_next_image()”.
  Refer to the grab_next_image_opencv sample code.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image_opencv sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.
  Refer to the grab_buffered_image_opencv sample code.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.7. grab_next_image

This sample code describes how to continuously capture images using the get_next_image function.

```
2 camera(s) found.
shape      : (540, 720, 3)
image data :
 [[  0   0   0]
 [  0   0   0]
 [  0   0   0]
 ...
 [  0 255 255]
 [  0 255 255]
 [  0 255 255]]
average    : 120.88888888888889
```

[Main functions of this sample code]

get_next_image()

[Remarks]

The get_next_image function waits until the image data that receives next is stored in the stream ring buffer inside this API, and get the image data when the store process is completed.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
  Refer to the grab_image_callback sample code.
- Using “get_next_image()”.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.
  Refer to the grab_buffered_image sample code.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

If capture images are got to continuously, the “set_trigger_mode” function is set to false, allowing to acquire images from the camera without waiting for a trigger.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.8. grab_next_image_using_trigger

This sample code describes how to capture an image using the get_next_image function.

```
2 camera(s) found.
------------------------------------------------------------------
Press '0' + 'Enter' key to issue "Software Trigger" and grab a frame.
Press '9' + 'Enter' key to quit application.
------------------------------------------------------------------
0
shape      : (540, 720, 3)
image data :
 [[  0   0   0]
 [  0   0   0]
 [  0   0   0]
 ...
 [  0 255 255]
 [  0 255 255]
 [  0 255 255]]
average    : 120.88888888888889
```

[Main functions of this sample code]

get_next_image()

[Remarks]

The get_next_image function waits until the image data that receives next is stored in the stream ring buffer inside this API, and get the image data when the store process is completed.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
  Refer to the grab_image_callback_using_trigger sample code.
- Using “get_next_image()”.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image_using_trigger sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.
  Refer to the grab_buffered_image_using_trigger sample code.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

The “set_trigger_mode” function can set both the software trigger and the hardware trigger.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.9. grab_next_image_opencv

This sample code describes how to continuously capture images using the get_next_image function and display images using the OpenCV. This image is test pattern (ColorBar).

[Main functions of this sample code]

get_next_image()

[Remarks]

The get_next_image function waits until the image data that receives next is stored in the stream ring buffer inside this API, and get the image data when the store process is completed.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with "set_callback_image_acquired()".
  Refer to the grab_image_callback_opencv sample code.
- Using "get_next_image()".
- Using "get_current_buffered_image()".
  Refer to the grab_current_image_opencv sample code.
- Using "get_current_buffer_index()、get_buffered_image()".
  Refer to the grab_buffered_image_opencv sample code.

Refer to 7.4 CameraStream class (Stream Control) of "pytelicam User Guide Eng.pdf" about for more information how to get images.

Refer to "pytelicam User Guide Eng.pdf" about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.10. grab_image_opencv-like

This sample code describes how to continuously capture images using the get_next_image function and display images using the OpenCV. It uses a sample class that has function like the VideoCapture of the OpenCV. This image is test pattern (ColorBar).

[Main functions of this sample code]

get_next_image()

[Remarks]

The get_next_image function waits until the image data that receives next is stored in the stream ring buffer inside this API, and get the image data when the store process is completed.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with "set_callback_image_acquired()".
  Refer to the grab_image_callback_opencv sample code.
- Using "get_next_image()".
- Using "get_current_buffered_image()".
  Refer to the grab_current_image_opencv sample code.
- Using "get_current_buffer_index()、get_buffered_image()".
  Refer to the grab_buffered_image_opencv sample code.

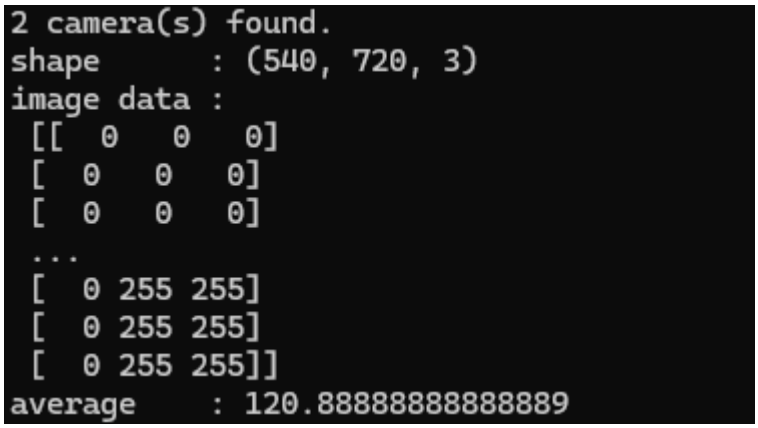
Refer to 7.4 CameraStream class (Stream Control) of "pytelicam User Guide Eng.pdf" about for more information how to get images.

Refer to "pytelicam User Guide Eng.pdf" about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.11. grab_current_image

This sample code describes how to continuously capture images using the get_current_buffered_image function.



[Main functions of this sample code]

get_current_buffered_image()

[Remarks]

The get_current_buffered_image function gets the latest stored image data from the stream ring buffer inside this API.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
  Refer to the grab_image_callback sample code.
- Using “get_next_image()”.
  Refer to the grab_next_image sample code.
- Using “get_current_buffered_image()”.
- Using “get_current_buffer_index()、get_buffered_image()”.
  Refer to the grab_buffered_image sample code.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

If capture images are got to continuously, the “set_trigger_mode” function is set to false, allowing to acquire images from the camera without waiting for a trigger.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.12. grab_current_image_using_trigger

This sample code describes how to capture an image using the get_current_buffered_image function.

```
2 camera(s) found.
------------------------------------------------------------------
Press '0' + 'Enter' key to issue "Software Trigger" and grab a frame.
Press '9' + 'Enter' key to quit application.
------------------------------------------------------------------
0
shape      : (540, 720, 3)
image data :
 [[  0   0   0]
 [  0   0   0]
 [  0   0   0]
 ...
 [  0 255 255]
 [  0 255 255]
 [  0 255 255]]
average    : 120.88888888888889
```

[Main functions of this sample code]

get_current_buffered_image()

[Remarks]

The get_current_buffered_image function gets the latest stored image data from the stream ring buffer inside this API.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
  Refer to the grab_image_callback_using_trigger sample code.
- Using “get_next_image()”.
  Refer to the grab_next_image_using_trigger sample code.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image_using_trigger sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

The “set_trigger_mode” function can set both the software trigger and the hardware trigger.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.13. grab_current_image_opencv

This sample code describes how to continuously capture images using the get_current_buffered_image function and display images using the OpenCV. This image is test pattern (ColorBar).

[Main functions of this sample code]

get_current_buffered_image()

[Remarks]

The get_current_buffer_image function gets the latest stored image data from the stream ring buffer inside this API.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
  Refer to the grab_image_callback_opencv sample code.
- Using “get_next_image()”.
  Refer to the grab_next_image_opencv sample code.
- Using “get_current_buffered_image()”.
- Using “get_current_buffer_index()、get_buffered_image()”.
  Refer to the grab_buffered_image_opencv sample code.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.14. grab_buffered_image

This sample code describes how to continuously capture images using the get_current_buffer_index function、the get_buffered_image function.

```
2 camera(s) found.
------------------------------------------------------------------
Press '0' + 'Enter' key to issue "Software Trigger" and grab a frame.
Press '9' + 'Enter' key to quit application.
------------------------------------------------------------------
0
shape      : (540, 720, 3)
image data :
 [[  0   0   0]
 [  0   0   0]
 [  0   0   0]
 ...
 [  0 255 255]
 [  0 255 255]
 [  0 255 255]]
average    : 120.88888888888889
```

[Main functions of this sample code]

get_current_buffer_index()、get_buffered_image()

[Remarks]

The get_current_buffer_index funtion and the get_buffered_image function can get the image data from any buffer(ImageData object) in the streaming buffer.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
  Refer to the grab_image_callback sample code.
- Using “get_next_image()”.
  Refer to the grab_next_image sample code.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

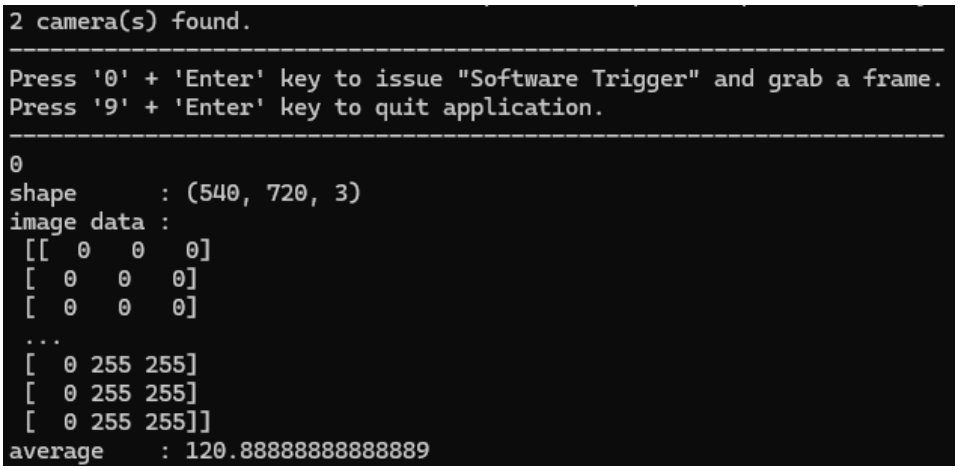
If capture images are got to continuously, the “set_trigger_mode” function is set to false, allowing to acquire images from the camera without waiting for a trigger.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.15. grab_buffered_image_using_trigger

This sample code describes how to capture an image using the get_current_buffer_index function、the get_buffered_image function.



[Main functions of this sample code]

get_current_buffer_index()、get_buffered_image()

[Remarks]

The get_current_buffer_index funtion and the get_buffered_image function can get the image data from any buffer(ImageData object) in the streaming buffer.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
  Refer to the grab_image_callback_using_trigger sample code.
- Using “get_next_image()”.
  Refer to the grab_next_image_using_trigger sample code.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image_using_trigger sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.

Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

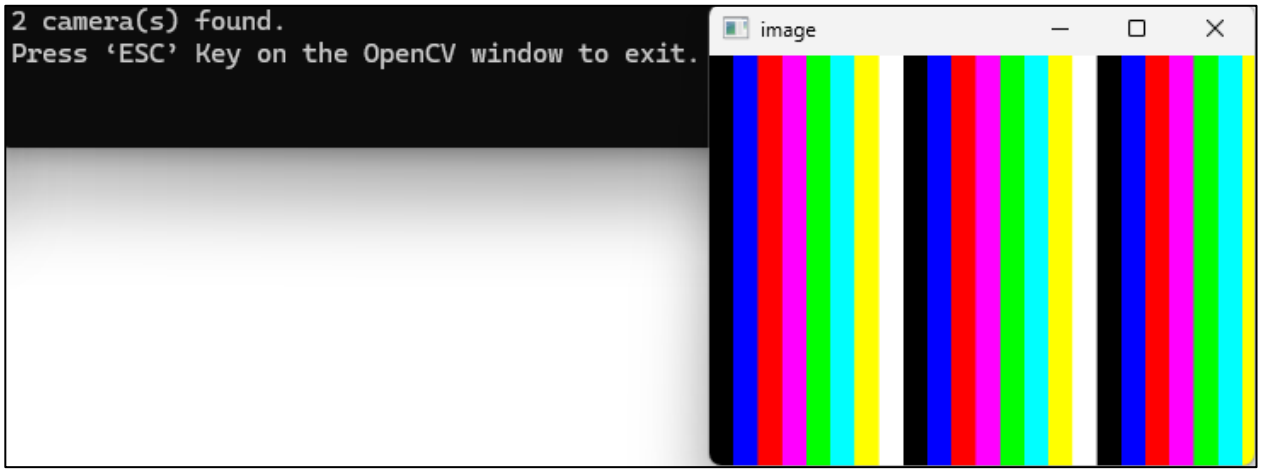
The “set_trigger_mode” function can set both the software trigger and the hardware trigger.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.16. grab_buffered_image_opencv

This sample code describes how to continuously capture images using the get_current_buffered_image, get_buffered_image function and display images using the OpenCV. This image is test pattern (ColorBar).



[Main functions of this sample code]

get_current_buffer_index()、get_buffered_image()

[Remarks]

The get_current_buffer_index funtion and the get_buffered_image function can get the image data from any buffer(ImageData object) in the streaming buffer.

There are ways to capture an image, including the method in this sample code.

- Using argument of callback function registered with “set_callback_image_acquired()”.
  Refer to the grab_image_callback_opencv sample code.
- Using “get_next_image()”.
  Refer to the grab_next_image_opencv sample code.
- Using “get_current_buffered_image()”.
  Refer to the grab_current_image_opencv sample code.
- Using “get_current_buffer_index()、get_buffered_image()”.

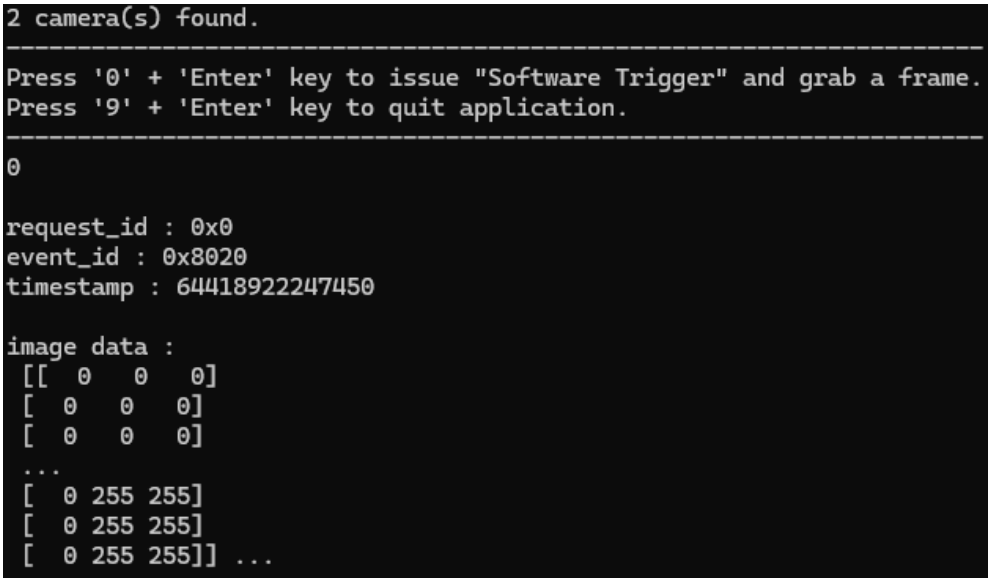
Refer to 7.4 CameraStream class (Stream Control) of “pytelicam User Guide Eng.pdf” about for more information how to get images.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.17. grab_camera_event

This sample code describes how to get camera event.



[Main functions of this sample code]

open()、activate()、get_event_data()、deactivate()

[Remarks]

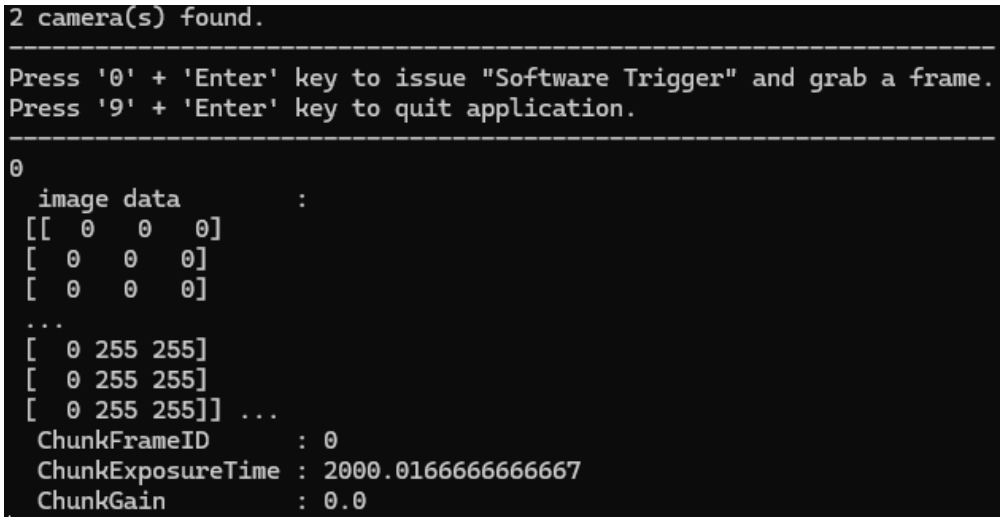
Get the FrameTrigger event and display the time stamp in this sample code.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.18. grab_chunk_data

This sample code describes how to use Chunk features.
Chunk features are used to acquire and set Chunk data (tagged blocks of data).
For details about Chunk features, refer to instruction manual of the camera.
Display the ColorBar of test pattern in the following image for example.



[Main functions of this sample code]

open()、activate()、get_event_data()、deactivate()

[Remarks]

Display and get FrameID, ExposureTime,
Gain, UserArea from Chunk data while an image is captured.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.19. get_camera_feature_list

This sample code describes how to display a list of names, types, access mode of features of a camera. Display items of enumeration when the type is it.

```
2 camera(s) found.

Root :
  DeviceControl : NodeType.Category, NodeAccessMode.ReadOnly
    DeviceVendorName : NodeType.String, NodeAccessMode.ReadOnly
    DeviceModelName : NodeType.String, NodeAccessMode.ReadOnly
    DeviceManufacturerInfo : NodeType.String, NodeAccessMode.ReadOnly
    DeviceVersion : NodeType.String, NodeAccessMode.ReadOnly
    DeviceFirmwareVersion : NodeType.String, NodeAccessMode.ReadOnly
    DeviceID : NodeType.String, NodeAccessMode.ReadOnly
    DeviceUserID : NodeType.String, NodeAccessMode.ReadAndWrite
  ImageFormatControl : NodeType.Category, NodeAccessMode.ReadOnly
    ImageFormatSelector : NodeType.Enumeration, NodeAccessMode.ReadAndWrite
      - Format2
      - Format1
      - Format0
    SensorWidth : NodeType.Integer, NodeAccessMode.ReadOnly
    SensorHeight : NodeType.Integer, NodeAccessMode.ReadOnly
    WidthMax : NodeType.Integer, NodeAccessMode.ReadOnly
    HeightMax : NodeType.Integer, NodeAccessMode.ReadOnly
    Width : NodeType.Integer, NodeAccessMode.ReadAndWrite
    Height : NodeType.Integer, NodeAccessMode.ReadAndWrite
    OffsetX : NodeType.Integer, NodeAccessMode.ReadAndWrite
    OffsetY : NodeType.Integer, NodeAccessMode.ReadAndWrite
    BinningHorizontal : NodeType.Integer, NodeAccessMode.ReadAndWrite
    BinningVertical : NodeType.Integer, NodeAccessMode.ReadAndWrite
```

[Main functions of this sample code]

get_available_feature_names()、get_node_type()、get_access_mode()、
get_available_enum_entry_names()

[Remarks]

Get names, types, access mode of camera features using the functions of GenApiWrapper class in this sample code.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

3.20. multi_camera_opencv

This sample code describes how to display images of multi cameras in a form using OpenCV. These images are test pattern. Left image is ColorBar and right image is GreyHorizontalRamp.

[Main functions of this sample code]

create_device_object()

[Remarks]

This sample code display images using OpenCV.

Refer to “pytelicam User Guide Eng.pdf” about for more information of features in this sample code.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

4. Others

4.1. Disclaimer

The disclaimer for the pytelicam follows the disclaimer for the TeliCamSDK.

The disclaimer of TeliCamSDK is described in another “License Agreement TeliCamSDK Eng.pdf”.
Make sure to read this Agreement carefully before using it.
Refer to TeliCamSDK installation folder/Licenses folder

4.2. License

The license for the pytelicam follows the license for the TeliCamSDK.

Microsoft, Windows, Windows XP, Windows Vista, Windows 7, Windows 8.1, Windows 10, Windows 11 and Visual C++ are the trademark or the registered trademark of Microsoft Corporation.
USB3 Vision and GigE Vision are trademark or registered trademark of AIA (Automated Imaging Association) of each company.
CoaXPress is registered trademark of JIIA (Japan Industrial Imaging Association).
GenICam is trademark of EMVA (European Machine Vision Association).
Furthermore, company name or product name might be trademark or registered trademark of each company.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/

4.3. Revision History

Date	Version	Description
2024/12/13	1.0.0	Created the initial version.

4.4. Inquiry

For frequently asked questions (FAQ) and answers about TeliCamSDK, GigE cameras, USB3 cameras, and CoaXPress cameras, please visit the "Support" - "Industrial Cameras FAQ" site on our website.

If you still cannot solve the problem, please contact us using the phone number or Inquiries form from "Contact Us" site on our website.

Copyright © 2024 Toshiba Teli Corporation, All rights reserved. https://www.toshiba-teli.co.jp/en/