pytelicam Samples

(Python) マニュアル

Version 1.0.0 (2024/12/13)

東芝テリー株式会社

改善の為予告なく変更することがありますので、最新の取扱説明書にて機能をご確認ください

DAA05080A

 https://www.toshiba-teli.co.jp/

目次

1. はじめに

本ドキュメントでは、pytelicam のサンプルコードについて説明します。サンプルコードは、pytelicam をインストールする事により利用可能となります。東芝テリーのホームページから pytelicam をダウンロードし、インストールしてください。

サンプルコードを理解するためには GenICam、GigE Vision、USB3 Vision、IIDC2 レジスタマップなどの知識が必要になる場合があります。これらの規格に関する詳細情報、最新情報はそれぞれの規格提供元のホームページを参照してください。

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. サンプルコードのファイルパス

サンプルコードは、pytelicam パッケージ内の samples フォルダに入っています。

pytelicam のインストール方法は、TeliCamSDK の「pytelicam User Guide Jpn.pdf」 を参照してください。

3. Python のサンプルコード

pytelicam は、python ユーザーアプリケーション実装の参考として以下の表に記載したサンプルコードを同梱しています。サンプルコードは順次追加予定です。

サンプル名	UI	機能
get_camera_information	CUI	カメラ情報の表示。
configure_camera_feature		CameraControl クラスの関数を使用したパラメータの取得と設定。
configure_camera_feature_using_genicam		GenApiWrapper クラスの関数を使用したパラメータの取得と設定。
grab_image_callback		コールバック関数を使用した画像の連続取込。
grab_image_callback_using_trigger		コールバック関数を使用した画像のソフトウェアトリガ取込。
grab_image_callback_opencv	GUI	コールバック関数を使用した画像の連続取込と OpenCV での画像表示。
grab_next_image	CUI	get_next_image 関数を使用した画像の連続取込。
grab_next_image_using_trigger		get_next_image 関数を使用した画像のソフトウェアトリガ取込。
grab_next_image_opencv	GUI	get_next_image 関数を使用した画像の連続取込と OpenCV での画像表示。
grab_image_opencv-like		get_next_image 関数を使用した画像の連続取込と OpenCV での画像表示。OpenCV の VideoCapture クラスと同様の機能を持つサンプルクラスを使用。
grab_current_image	CUI	get_current_buffered_image関数を使用した画像の連続取込。
grab_current_image_using_trigger		get_current_buffered_image 関数を使用した画像のソフトウェアトリガ取込。
grab_current_image_opencv	GUI	get_current_buffered_image 関数を使用した画像の連続取込と OpenCV での画像表示。
grab_buffered_image	CUI	get_current_buffer_index 関数, get_buffered_image 関数を使用した画像の連続取込。
grab_buffered_image_using_trigger		get_current_buffer_index 関数, get_buffered_image 関数を使用した画像のソフトウェアトリガ取込。
grab_buffered_image_opencv	GUI	get_current_buffer_index 関数, get_buffered_image 関数を使用した画像の連続取込と OpenCV での画像表示。
grab_camera_event	CUI	カメライベント取得。
grab_chunk_data		カメラのチャンクデータの取得。
get_camera_feature_list		カメラの機能一覧の表示。
multi_camera_opencv	GUI	複数カメラの画像を opencv で表示。

3.1. get_camera_information

このサンプルコードは、接続している全てのカメラの情報を表示します。

```
<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
```

[使用例の主な関数]

get_camera_information()

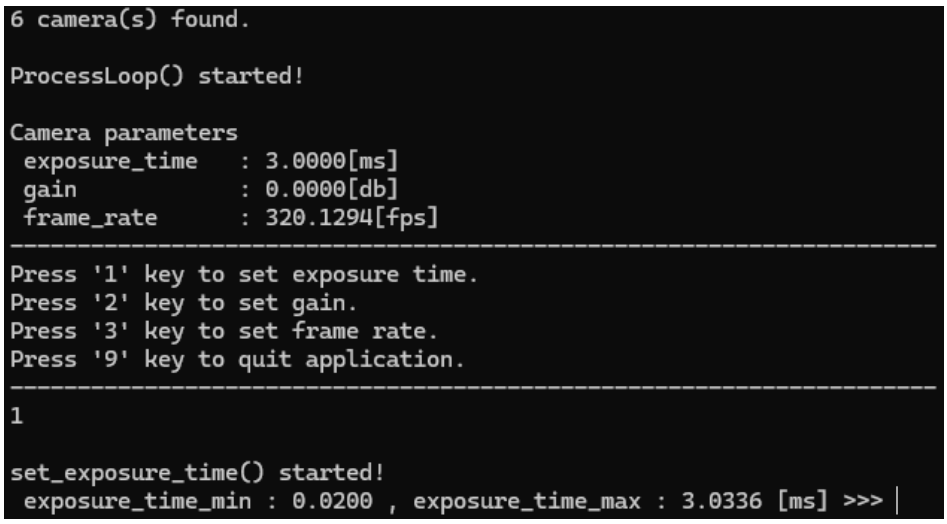
[備考]

get_camera_information 関数は、検出したカメラの情報を取得します。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.2. configure_camera_feature

このサンプルコードは、CameraControl クラスの関数を使用してカメラのパラメータを取得・設定する方法を示します。



[使用例の主な関数]

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()

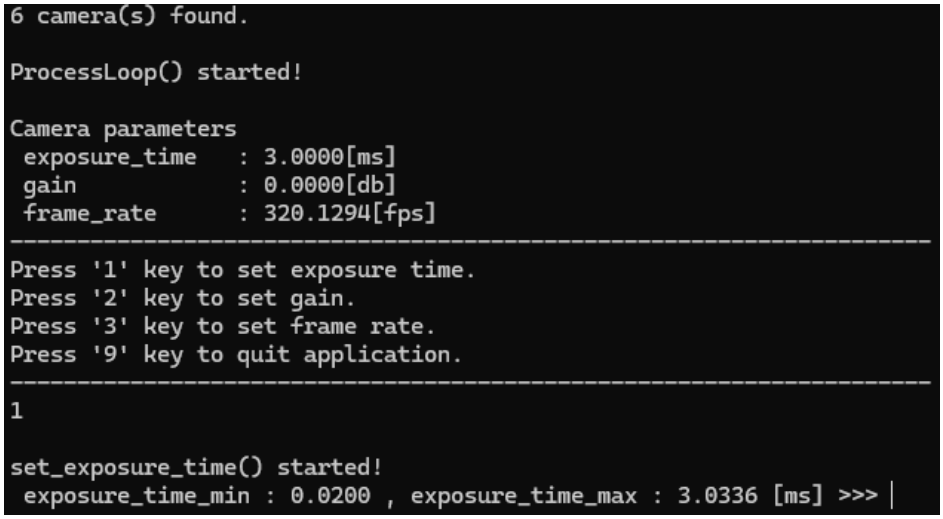
[備考]

CameraControl クラスの関数は、GenApiWrapper クラスの関数を使用した場合に比べて高速に処理できます。
ただし、すべてのカメラの機能に対して CameraControl クラスの関数が用意されているわけではありません。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.3. configure_camera_feature_using_genicam

このサンプルコードは、GenApiWrapper クラスの関数を使用してカメラのパラメータを取得・設定する方法を示します。



[使用例の主な関数]

get_float_min()、get_float_max()、set_float_value()

[備考]

CameraControl クラスの関数が用意されていないカメラの機能に対しても実行することができます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.4. grab_image_callback

このサンプルコードは、コールバック関数を使用した画像の連続取り込み方法を示します。

```
2 camera(s) found.
------------------------------------------------------------------------
Press '0' + 'Enter' key to grab frames.
Press '9' + 'Enter' key to quit application.
------------------------------------------------------------------------
0
shape      : (540, 720, 3)
image data :
 [[186 255 255]
 [188 255 255]
 [190 255 255]
 ...
 [137 255 200]
 [136 255 199]
 [136 255 198]]
average    : 102.90358281893005
```

[使用例の主な関数]

set_callback_image_acquired()、set_callback_image_error()、set_callback_buffer_busy()

[備考]

set_callback_image_acquired 関数は、コールバック関数を登録できます。登録したコールバック関数は、新しい画像データが本 API 内部のストリームリングバッファに格納されたときに呼び出されます。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類あります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

◆get_next_image()を使用する方法

使用方法は grab_next_image サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

使用方法は grab_current_image サンプルコードを参照してください。

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image サンプルコードを参照してください。

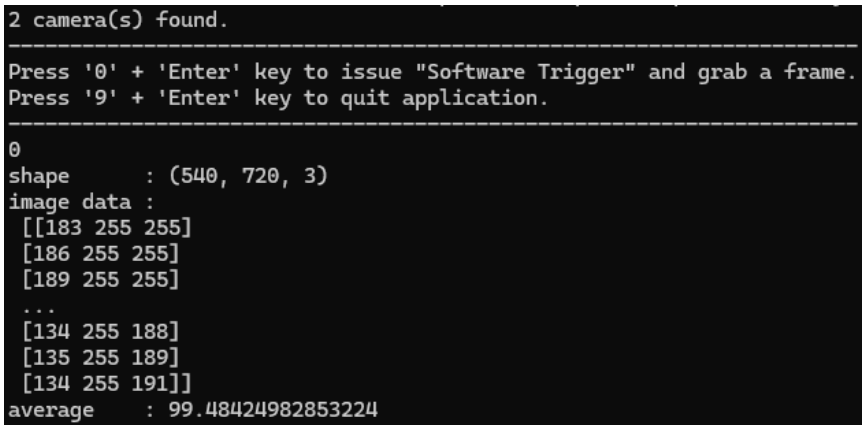
画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

画像を連続で取り込む場合、set_trigger_mode 関数を false に設定することでトリガを待たずにカメラから出力された画像を取得できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.5. grab_image_callback_using_trigger

このサンプルコードは、コールバック関数を使用した画像の取り込み方法を示します。



[使用例の主な関数]

set_callback_image_acquired()、set_callback_image_error()、set_callback_buffer_busy()

[備考]

set_callback_image_acquired 関数は、コールバック関数を登録できます。登録したコールバック関数は、新しい画像データが本 API 内部のストリームリングバッファに格納されたときに呼び出されます。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

◆get_next_image()を使用する方法

使用方法は grab_next_image_using_trigger サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

使用方法は grab_current_image_using_trigger サンプルコードを参照してください。

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image_using_trigger サンプルコードを参照してください。

画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

set_trigger_source()関数は、ソフトウェアトリガとハードウェアトリガの両方を設定できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.6. grab_image_callback_opencv

コールバック関数を使用した画像の連続取込と OpenCV で画像を表示する方法を示します。ここでは、例としてカメラのテストパターン(ColorBar)を表示しています。

[使用例の主な関数]

set_callback_image_acquired()、set_callback_image_error()、set_callback_buffer_busy()

[備考]

set_callback_image_acquired 関数は、コールバック関数を登録できます。登録したコールバック関数は、新しい画像データが本 API 内部のストリームリングバッファに格納されたときに呼び出されます。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

◆get_next_image()を使用する方法

使用方法は grab_next_image サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

使用方法は grab_current_image サンプルコードを参照してください。

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image サンプルコードを参照してください。

画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.7. grab_next_image

このサンプルコードは、get_next_image 関数を使用した画像の連続取り込み方法を示します。

```
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
```

[使用例の主な関数]

get_next_image()

[備考]

get_next_image 関数は、次に到着する画像データが本 API 内部のストリームリングバッファに格納されるまで待機し、格納処理が完了したら画像データを取得します。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback サンプルコードを参照してください。

◆get_next_image()を使用する方法

◆get_current_buffered_image()を使用する方法

使用方法は grab_current_image サンプルコードを参照してください。

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image サンプルコードを参照してください。

画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

画像を連続で取り込む場合、set_trigger_mode 関数を false に設定することでトリガを待たずにカメラから出力された画像を取得できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.8. grab_next_image_using_trigger

このサンプルコードは、get_next_image 関数を使用した画像の取り込み方法を示します。

```
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
```

[使用例の主な関数]

get_next_image()

[備考]

get_next_image 関数は、コールバック関数を登録できます。登録したコールバック関数は、新しい画像データが本 API 内部のストリームリングバッファに格納されたときに呼び出されます。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback_using_trigger サンプルコードを参照してください。

◆get_next_image()を使用する方法

◆get_current_buffered_image()を使用する方法

使用方法は grab_current_image_using_trigger サンプルコードを参照してください。

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image_using_trigger サンプルコードを参照してください。

画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

set_trigger_source()関数は、ソフトウェアトリガとハードウェアトリガの両方を設定できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.9. grab_next_image_opencv

このサンプルコードは、get_next_image 関数を使用した画像の連続取込と OpenCV で画像を表示する方法を示します。ここでは、例としてカメラのテストパターン(ColorBar)を表示しています。

[使用例の主な関数]

get_next_image()

[備考]

get_next_image 関数は、次に到着する画像データが本 API 内部のストリームリングバッファに格納されるまで待機し、格納処理が完了したら画像データを取得します。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback_opencv サンプルコードを参照してください。

◆get_next_image()を使用する方法

◆get_current_buffered_image()を使用する方法

使用方法は grab_current_image_opencv サンプルコードを参照してください。

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image_opencv サンプルコードを参照してください。

画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

画像を連続で取り込む場合、set_trigger_mode 関数を false に設定することでトリガを待たずにカメラから出力された画像を取得できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.10. grab_image_opencv-like

このサンプルコードは、get_next_image 関数を使用した画像の連続取込と OpenCV で画像を表示する方法を示します。OpenCV の VideoCapture クラスと同様の機能を持つサンプルクラスを使用します。ここでは、例としてカメラのテストパターン(ColorBar)を表示しています。

[使用例の主な関数]

get_next_image()

[備考]

get_next_image 関数は、次に到着する画像データが本 API 内部のストリームリングバッファに格納されるまで待機し、格納処理が完了したら画像データを取得します。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback_opencv サンプルコードを参照してください。

◆get_next_image()を使用する方法

◆get_current_buffered_image()を使用する方法

使用方法は grab_current_image_opencv サンプルコードを参照してください。

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image_opencv サンプルコードを参照してください。

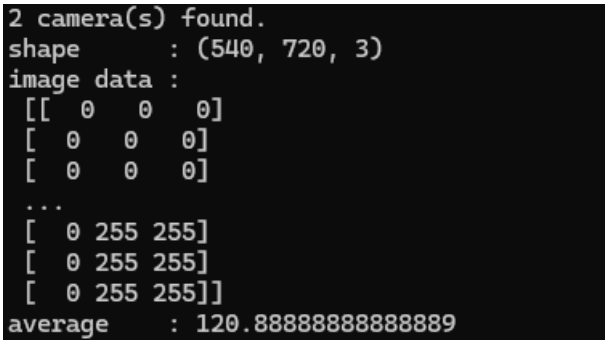
画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

画像を連続で取り込む場合、set_trigger_mode 関数を false に設定することでトリガを待たずにカメラから出力された画像を取得できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.11. grab_current_image

このサンプルコードは、get_current_buffered_image 関数を使用した画像の連続取り込み方法を示します。



[使用例の主な関数]

get_current_buffered_image()

[備考]

get_current_buffered_image 関数は、本 API 内部のストリームリングバッファから格納済の最新画像データを取得します。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback サンプルコードを参照してください。

◆get_next_image()を使用する方法

使用方法は grab_next_image サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image サンプルコードを参照してください。

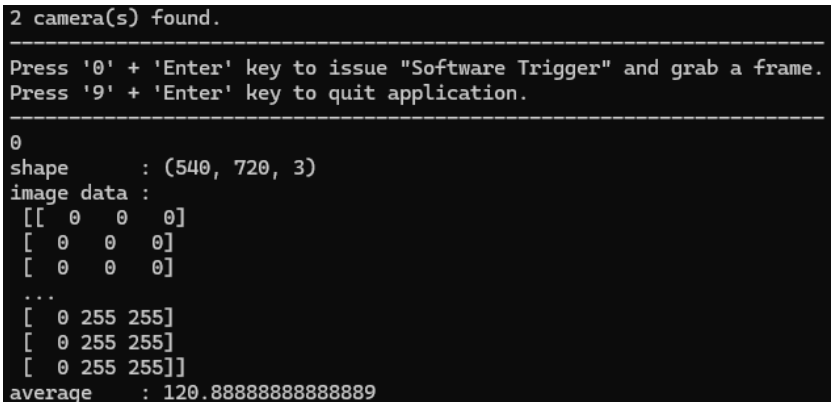
画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

画像を連続で取り込む場合、set_trigger_mode 関数を false に設定することでトリガを待たずにカメラから出力された画像を取得できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.12. grab_current_image_using_trigger

このサンプルコードは、get_current_buffered_image 関数を使用した画像の連続取り込み方法を示します。



[使用例の主な関数]

get_current_buffered_image()

[備考]

get_current_buffered_image 関数は、本 API 内部のストリームリングバッファから格納済の最新画像データを取得します。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback_using_trigger サンプルコードを参照してください。

◆get_next_image()を使用する方法

使用方法は grab_next_image_using_trigger サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image_using_trigger サンプルコードを参照してください。

画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

set_trigger_source()関数は、ソフトウェアトリガとハードウェアトリガの両方を設定できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.13. grab_current_image_opencv

このサンプルコードは、get_current_buffered_image 関数を使用した画像の連続取込と OpenCV で画像を表示する方法を示します。ここでは、例としてカメラのテストパターン(ColorBar)を表示しています。

[使用例の主な関数]

get_current_buffered_image()

[備考]

get_current_buffered_image 関数は、本 API 内部のストリームリングバッファから格納済の最新画像データを取得します。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback_opencv サンプルコードを参照してください。

◆get_next_image()を使用する方法

使用方法は grab_next_image_opencv サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image_opencv サンプルコードを参照してください。

画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

画像を連続で取り込む場合、set_trigger_mode 関数を false に設定することでトリガを待たずにカメラから出力された画像を取得できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.14. grab_buffered_image

このサンプルコードは、get_current_buffer_index 関数、get_buffered_image 関数を使用した画像の連続取り込み方法を示します。

```
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
```

[使用例の主な関数]

get_current_buffer_index()、get_buffered_image()

[備考]

get_current_buffer_index()関数、get_buffered_image()関数を使用することで、ストリームリングバッファ内にある任意のインデックスの画像データを取得できます。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback サンプルコードを参照してください。

◆get_next_image()を使用する方法

使用方法は grab_next_image サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image サンプルコードを参照してください。

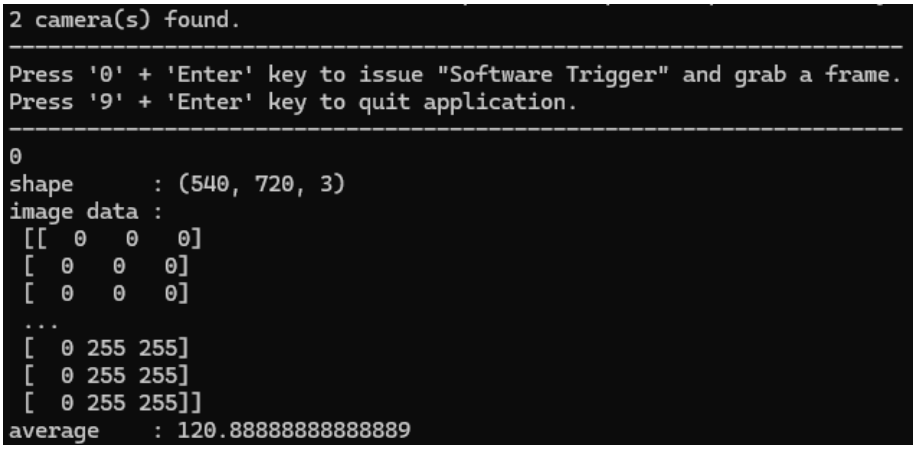
画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

画像を連続で取り込む場合、set_trigger_mode 関数を false に設定することでトリガを待たずにカメラから出力された画像を取得できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.15. grab_buffered_image_using_trigger

このサンプルコードは、get_current_buffer_index 関数、get_buffered_image 関数を使用した画像の取り込み方法を示します。



[使用例の主な関数]

get_current_buffer_index()、get_buffered_image()

[備考]

get_current_buffer_index()関数、get_buffered_image()関数を使用することで、ストリームリングバッファ内にある任意のインデックスの画像データを取得できます。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback_using_trigger サンプルコードを参照してください。

◆get_next_image()を使用する方法

使用方法は grab_next_image_using_trigger サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

使用方法は grab_current_image_using_trigger サンプルコードを参照してください。

◆get_current_buffer_index()、get_buffered_image()を使用する方法

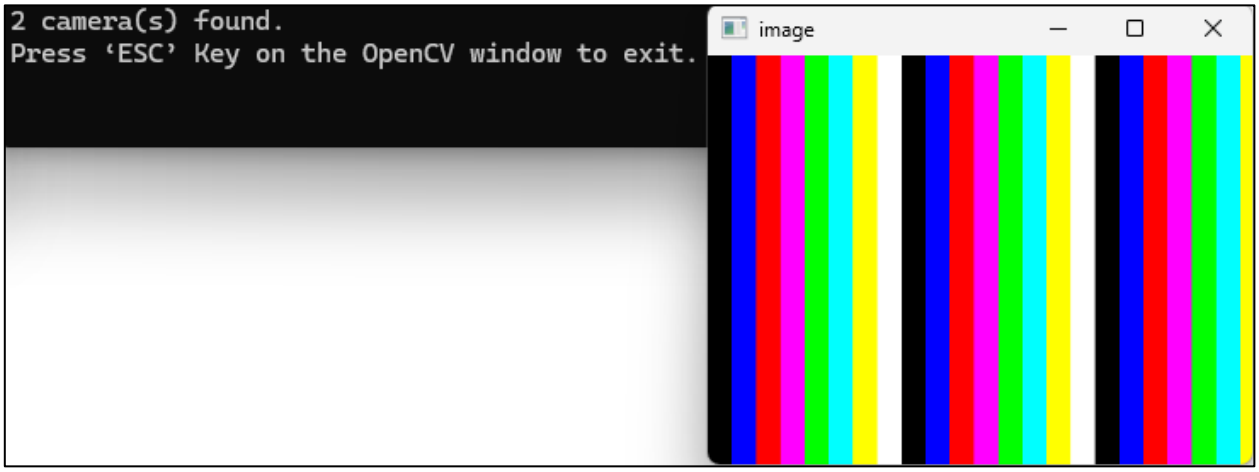
画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

set_trigger_source()関数は、ソフトウェアトリガとハードウェアトリガの両方を設定できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.16. grab_buffered_image_opencv

このサンプルコードは、get_current_buffer_index 関数、get_buffered_image 関数を使用した画像の連続取込と OpenCV で画像を表示する方法を示します。ここでは、例としてカメラのテストパターン(ColorBar)を表示しています。



[使用例の主な関数]

get_current_buffer_index()、get_buffered_image()

[備考]

get_current_buffer_index()関数、get_buffered_image()関数を使用することで、ストリームリングバッファ内にある任意のインデックスの画像データを取得できます。

画像の取り込み方法は、このサンプルコードの方法を含めて以下の種類があります。

◆set_callback_image_acquired()で登録したコールバック関数の引数を使用した方法

使用方法は grab_image_callback_opencv サンプルコードを参照してください。

◆get_next_image()を使用する方法

使用方法は grab_next_image_opencv サンプルコードを参照してください。

◆get_current_buffered_image()を使用する方法

◆get_current_buffer_index()、get_buffered_image()を使用する方法

使用方法は grab_buffered_image_opencv サンプルコードを参照してください。

画像取り込み方法の詳細は、「pytelicam User Guide Jpn.pdf」の 7.4CameraStream クラス(ストリーム制御)を参照してください。

画像を連続で取り込む場合、set_trigger_mode 関数を false に設定することでトリガを待たずにカメラから出力された画像を取得できます。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.17. grab_camera_event

このサンプルコードは、カメライベントの取得方法を示します。

```
2 camera(s) found.
------------------------------------------------------------------------
Press '0' + 'Enter' key to issue "Software Trigger" and grab a frame.
Press '9' + 'Enter' key to quit application.
------------------------------------------------------------------------
0

request_id : 0x0
event_id : 0x8020
timestamp : 64418922247450

image data :
 [[  0   0   0]
 [  0   0   0]
 [  0   0   0]
 ...
 [  0 255 255]
 [  0 255 255]
 [  0 255 255]] ...
```

[使用例の主な関数]

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

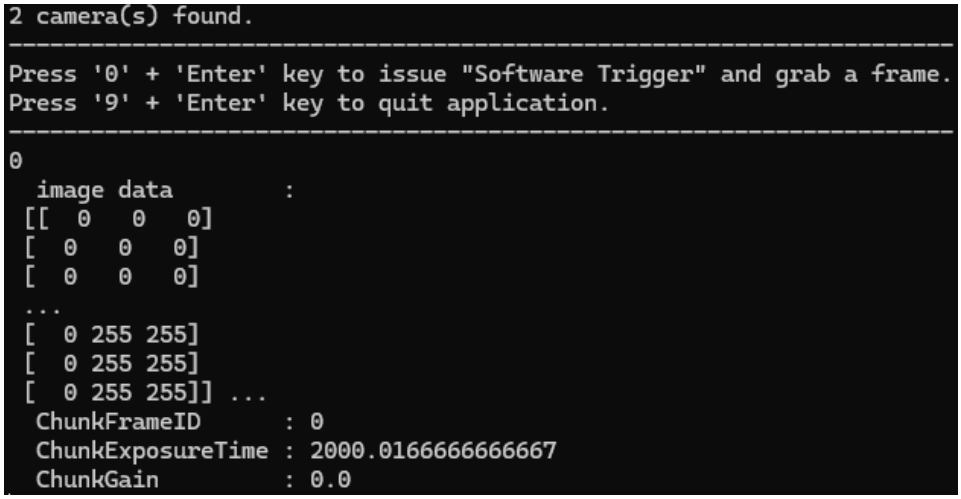
[備考]

このサンプルは、FrameTrigger イベントを取得してタイムスタンプを表示します。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.18. grab_chunk_data

このサンプルコードは、カメラのチャンク機能の使用方法を示しています。
チャンク機能を使用するとチャンクデータ(画像データ毎に付加されたタグ情報)の取得と設定を行えます。カメラのチャンク機能に関する詳しい説明は、使用するカメラの取扱説明書をご覧ください。



[使用例の主な関数]

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

[備考]

このサンプルは、FrameTrigger イベントを取得してタイムスタンプを表示します。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.19. get_camera_feature_list

このサンプルコードは、1 台のカメラの機能の名前と属性、アクセスモードを一覧で表示します。属性が Enumeration の場合はその項目も表示します。

```
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
```

[使用例の主な関数]

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

[備考]

このサンプルコードは、GenApiWrapper クラスの関数を使用して機能の名前と属性、アクセスモードを取得しています。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

3.20. multi_camera_opencv

このサンプルコードは、複数のカメラの画像を opencv で表示する方法を示しています。ここでは、例としてカメラのテストパターンを表示しています。左画像は ColorBar、右画像は GreyHorizontalRamp です。

[使用例の主な関数]

create_device_object()

[備考]

このサンプルコードは、OpenCV を利用して画像を表示します。

サンプルコード内の関数の詳細については、「pytelicam User Guide Jpn.pdf」を参照してください。

4. その他

4.1. 免責事項

pytelicam ライブラリの免責事項は、TeliCamSDK の免責事項に準じます。

TeliCamSDK の免責事項は、TeliCamSDK に付属の”License Agreement TeliCamSDK Jpn.pdf”に記載されています。　必ずご一読の上、ご利用されますようお願い致します。
[TeliCamSDK インストールフォルダ]\Licenses フォルダを参照ください。

4.2. ライセンス

pytelicam ライブラリのライセンスは、TeliCamSDK のライセンスに準じます。
Microsoft、Windows、Windows XP、Windows Vista、Windows 7、Windows 8.1、Windows 10、Windows 11 及び Visual C++ は、Microsoft 社の商標もしくは登録商標です。
GigE Vision は、AIA (Automated Imaging Association)の各国における商標または登録商標です。
USB3 Vision は、AIA (Automated Imaging Association)の各国における商標または登録商標です。
GenICam は、EMVA (European Machine Vision association)の各国における商標または登録商標です。
CoaXPress は、JIIA (Japan Industrial Imaging Association)の登録商標です。
その他、本ドキュメントに記載の会社名、団体名、製品名、規格名等の名称は、各社各団体における商標または登録商標です。

4.3. 改定履歴

Date	Version	内容
2024/12/13	1.0.0	初版

4.4. お問い合わせ

TeliCamSDK ならびにGigE カメラ、USB3 カメラ、CoaXPress カメラに関するよくあるご質問とその回答(FAQ)は、弊社ホームページの「サポート」 ー 「産業カメラに関するFAQ」サイトをご利用ください。

それでも解決できない場合は、弊社ホームページの「お問い合わせ」サイトに記載の電話番号またはメールフォームにてご連絡ください。