Helpdesk XIMEA

Python API

Support SK - 2023-10-19 - in xiApiPython

https://www.ximea.com/support/wiki/apis/python

xiAPI Python¶

Applications in Python can access XIMEA cameras using xiAPI Python interface.
This API creates an interface with which you can fully utilize the features and capabilities of XIMEA cameras.
This Python interface supports various Python packages like OpenCV or Pillow.

Documentation¶

xiAPI Python Manual

Supported platforms¶

- Windows
- Linux
- MacOS

Installation instructions¶

- Installation for Windows
- Installation for Linux
- Installation for MacOS

Sample code:¶

The Ximea Python package may be used in a script in the following way:

```
from ximea import xiapi

#create instance for first connected camera
cam = xiapi.Camera()

#start communication
#to open specific device, use:
#cam.open_device_by_SN('41305651')
```

```
#(open by serial number)
print('Opening first camera...')
cam.open_device()

#settings
cam.set_exposure(10000)
print('Exposure was set to %i us' %cam.get_exposure())

#create instance of Image to store image data and metadata
img = xiapi.Image()

#start data acquisition
print('Starting data acquisition...')
cam.start_acquisition()

for i in range(10):
    #get data and pass them from camera to img
    cam.get_image(img)

    #get raw data from camera
    #for Python2.x function returns string
    #for Python3.x function returns bytes
    data_raw = img.get_image_data_raw()

    #transform data to list
    data = list(data_raw)

    #print image data and metadata
    print('Image number: ' + str(i))
    print('Image width (pixels):  ' + str(img.width))
    print('Image height (pixels): ' + str(img.height))
    print('First 10 pixels: ' + str(data[:10]))
    print('\n')

#stop data acquisition
print('Stopping acquisition...')
cam.stop_acquisition()

#stop communication
cam.close_device()
```

```
print('Done.')
```

XIMEA Python syntax¶

The various xiAPI parameters can be set or readout using XIMEA Python functions which are generated from the constant string definitions of the xiAPI parameters.

Enumerator parameters¶

Prefixes *get_* and *set_* are added to constant string definition the following way:
For parameter XI_PRM_ACQ_TIMING_MODE, **acq_timing_mode** is the constant string, therefor Python XIMEA API contains functions *get_acq_timing_mode()* and *set_acq_timing_mode()*:

```
cam.set_acq_timing_mode('XI_ACQ_TIMING_MODE_FRAME_RATE')
mode_used = cam.get_acq_timing_mode()
```

The returned value is string, so checking the retured value could be compared with another string:

```
if mode_used == 'XI_ACQ_TIMING_MODE_FRAME_RATE':
    print('Mode is XI_ACQ_TIMING_MODE_FRAME_RATE')
else:
    print('Mode is not XI_ACQ_TIMING_MODE_FRAME_RATE')
```

Enumerator inputs must be specific strings. Input in form of integer value that these strings represent will result in errors:

```
camera.set_gpo_selector('XI_GPO_PORT1') #will work correctly
camera.set_gpo_selector(1) #will result in error
```

Nonboolean parameters¶

Prefixes *get_* and *set_* are added the same way as for enumerator parameters:
E.g. for parameter XI_PRM_EXPOSURE, **"exposure"** is the constant string, therefore Python XIMEA API contains functions *get_exposure()* and *set_exposure()*:

```
cam.set_exposure(10000)
print('Current exposure is %s us.' %cam.get_exposure())
```

and the output will be:

```
Current exposure is 10000 us.
```

For setting the parameter XI_PRM_FRAMERATE to 10, the commands will be:

```
cam.set_acq_timing_mode('XI_ACQ_TIMING_MODE_FRAME_RATE')
cam.set_framerate(10)
```

Note, that the value *XI_ACQ_TIMING_MODE_FRAME_RATE* works only with MQ and MD cameras. More information regarding setting the fixed framerate can be found in xiAPI Manual and in our knowledge base article Frame Rate Control.
For getting API parameter modifiers commands will be:

```
print('The maximal width of this camera is %i.'
%cam.get_width_maximum())
print('The minimal width of this camera is %i.'
%cam.get_width_minimum())
print('The increment of the width of this camera is %i.'
%cam.get_width_increment())
```

and the output for e.g. MQ013MG-ON camera will be:

```
The maximal width of this camera is 1280.
The minimal width of this camera is 16.
The increment of the width of this camera is 16.
```

Boolean parameters¶

Same logic is used for boolean parameters, where prefixes *is_*, *enable_* and *disable_* are used the following way:
For parameter XI_PRM_HORIZONTAL_FLIP, **"horizontal_flip"** is the constant string, therefor Python XIMEA API contains functions *is_horizontal_flip()*, *enable_horizontal_flip()* and *disable_horizontal_flip()*:

```
print('Is horizontal flip enabled?')
print(cam.is_horizontal_flip())

print('Enabling horizontal flip...')
cam.enable_horizontal_flip()
print(cam.is_horizontal_flip())

print('Disabling horizontal flip...')
cam.disable_horizontal_flip()
print(cam.is_horizontal_flip())
```

and the output will be:

```
Is horizontal flip enabled?
False.
Enabling horizontal flip...
True.
Disabling horizontal flip...
False.
```

Common interface¶

For setting or getting parameters can be used *get_param(arg)* and *set_param(arg)* functions, where as **arg** is used the constant string of specific parameter the following way:

```
cam.set_param("exposure",10000)
// is the same as
cam.set_exposure(10000)
```

```
cam.get_param("width:max")
// is the same as
cam.get_width_maximum()
```

Both approaches are valid.

Example applications¶

There are several applications, that may be found in the xiApiPython examples/ folder. Here you may find various examples for xiApiPython, OpenCV, or Pillow packages.

Note¶

The Python v3. supports examples only from *pillow* and *various* folder.