Python 接口使用

本手册描述了 JHSM 系列工业相机在 Python 语言环境中的调用方式。

JHSM 系列工业相机提供 SDK 可以在 Python 环境中通过 ctypes 方式调用 API 获取图像或者参数。

1 准备工作

1）Windows 将 JHCap2.dll 拷贝到 PATH 路径下

32 位 dll 在 32 位系统或者 64 位 dll 在 64 位系统位置：c:\windows\system32

32 位 dll 在 64 位系统位置：C:\windows\syswow64

2）Ubuntu 系统运行 install 后 libJHCap.so 将拷贝到系统目录下。

3）安装 Python 组件，需要用到 numpy 和 opencv 组件

```
Windows：
pip install opencv-python

Linux：
sudo apt-get install python-opencv

Test installation
   python
   >>>import cv2
   >>>cv2.__version__
   ’3.2.0’
```

4）API 文档请参考《SDK 开发说明书》，函数的常数定义见 const.txt。

5）

2 Python 代码 采集单张彩色图片并显示

```
>>> import numpy as np
>>> import cv2
>>> from ctypes import *
#call API to query image
>>> dll=cdll.LoadLibrary("JHCap2.dll") #在 Linux 下"JHCap2.dll"改为"libJHCap.so"
>>> dll.CameraInit(0)
>>> dll.CameraSetResolution(0, 0, 0, 0)
>>> buflen = c_int()
>>> width = c_int()
>>> height = c_int()
>>> dll.CameraGetImageSize(0, byref(width), byref(height))
>>> dll.CameraGetImageBufferSize(0, byref(buflen), 0x4)
>>> inbuf = create_string_buffer(buflen.value)
>>> dll.CameraQueryImage(0, inbuf, byref(buflen), 0x104)
#using opencv to display the buffer image
>>> arr= np.frombuffer(inbuf, np.uint8)
>>> img=np.reshape(arr, (height.value, width.value, 3))
>>> cv2.namedWindow(″s″)
>>> cv2.imshow(″s″, img)
>>> key=cv2.waitKey(0)
>>> if key == 27:
>>>     cv2.destroyWindow(″s″)
>>>
```

*以上代码在 Python3.5 64bit Windows 系统中测试通过.

可以直接复制的代码

1）显示单张图像

```
import numpy as np
import cv2
from ctypes import *
#call API to query image
dll=cdll.LoadLibrary("JHCap2.dll")
dll.CameraInit(0)
dll.CameraSetResolution(0, 0, 0, 0)
dll.CameraSetContrast.argtypes = [ c_int, c_double ]
dll.CameraSetContrast(0, 1.15)
buflen = c_int()
width = c_int()
height = c_int()
dll.CameraGetImageSize(0, byref(width), byref(height))
dll.CameraGetImageBufferSize(0, byref(buflen), 0x4)
inbuf = create_string_buffer(buflen.value)
dll.CameraQueryImage(0, inbuf, byref(buflen), 0x104)
#using opencv to display the buffer image
arr= np.frombuffer(inbuf, np.uint8)
```

```
img=np.reshape(arr, (height.value, width.value, 3))
cv2.namedWindow("s")
cv2.imshow("s", img)
key=cv2.waitKey(0)
if key == 27:           #press ESC on image window to exit
    cv2.destroyWindow("s")
```

2）循环显示图像

```
import numpy as np
import cv2
from ctypes import *
#call API to query image
dll=cdll.LoadLibrary("JHCap2.dll")
dll.CameraInit(0)
dll.CameraSetResolution(0, 0, 0, 0)
dll.CameraSetContrast.argtypes = [ c_int, c_double ]
dll.CameraSetContrast(0, 1.15)
buflen = c_int()
width = c_int()
height = c_int()
dll.CameraGetImageSize(0, byref(width), byref(height))
dll.CameraGetImageBufferSize(0, byref(buflen), 0x4)
inbuf = create_string_buffer(buflen.value)

cv2.namedWindow("s")
while 1:
    dll.CameraQueryImage(0, inbuf, byref(buflen), 0x104)
    arr= np.frombuffer(inbuf, np.uint8)
    img=np.reshape(arr, (height.value, width.value, 3))
    cv2.imshow("s", img)
    key=cv2.waitKey(60)          #change parameter according to frame rate, wait time = 1000/fps
    if key == 27:                #press ESC on image window to terminate the loop
        break
```

```
cv2.destroyWindow("s")
```

3) Ubuntu 下循环显示图片

```
import numpy as np
import cv2
from ctypes import *
#call API to query image
dll=cdll.LoadLibrary("libJHCap.so")
dll.CameraInit(0)
dll.CameraSetResolution(0, 0, 0, 0)
dll.CameraSetContrast.argtypes = [ c_int, c_double ]
dll.CameraSetContrast(0, 1.15)
buflen = c_int()
width = c_int()
height = c_int()
dll.CameraGetImageSize(0, byref(width), byref(height))
dll.CameraGetImageBufferSize(0, byref(buflen), 0x4)
inbuf = create_string_buffer(buflen.value)
cv2.namedWindow("s")
while 1:
    dll.CameraQueryImage(0, inbuf, byref(buflen), 0x104)
    arr= np.frombuffer(inbuf, np.uint8)
    img=np.reshape(arr, (height.value, width.value, 3))
    cv2.imshow("s", img)
    key=cv2.waitKey(33) #change parameter according to frame rate, wait time = 1000/fps
    if (key&0xff) == 27: #press ESC on image window to terminate the loop
        break
cv2.destroyWindow("s")
```