

pyrealsense2 frame_metadata_value class (frame metadata) (frame metadata refers to a set of read-only properties that may be exposed for each individual frame)

[other](#)

2020-01-17 23:22:02

Reads: 0



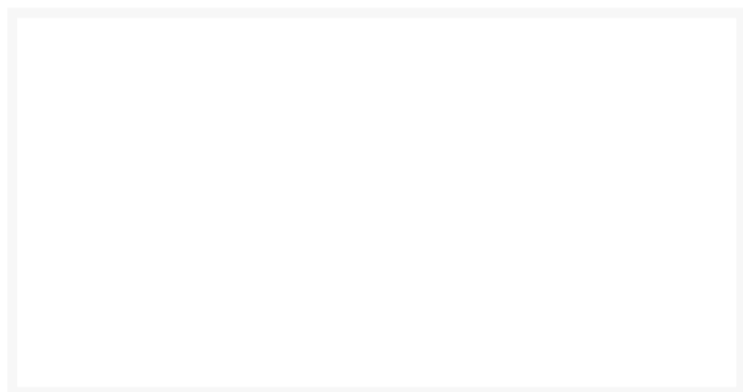
FREE CANCELLATION

MELIÁ SALI

From €26

Article Directory

- [from frame_metadata_value.py](#)
- [problem](#)



Meliá Alicante

广告 Alicante

Melia

BOOK >

from frame_metadata_value.py



[Report this ad](#)

```

"""
    Library for accessing Intel RealSenseTM cameras
"""

# imports
import pybind11_builtins as __pybind11_builtins

class frame_metadata_value(__pybind11_builtins.pybind11_object):
    """ Per-Frame-Metadata is the set of read-only properties that might be exposed for each individual frame.
    每帧元数据是可能针对每个单独的帧公开的一组只读属性 """
    def __eq__(self, *args, **kwargs): # real signature unknown; restored from __doc__
        """
        __eq__(*args, **kwargs)
        Overloaded function.

        1. __eq__(self: pyrealsense2.pyrealsense2.frame_metadata_value, arg0:
        pyrealsense2.pyrealsense2.frame_metadata_value) -> bool

        2. __eq__(self: pyrealsense2.pyrealsense2.frame_metadata_value, arg0: int) -> bool
        """
        pass

    def __getstate__(self): # real signature unknown; restored from __doc__
        """ __getstate__(self: pyrealsense2.pyrealsense2.frame_metadata_value) -> tuple """
        return ()

    def __hash__(self): # real signature unknown; restored from __doc__
        """ __hash__(self: pyrealsense2.pyrealsense2.frame_metadata_value) -> int """
        return 0

    def __init__(self, arg0): # real signature unknown; restored from __doc__
        """ __init__(self: pyrealsense2.pyrealsense2.frame_metadata_value, arg0: int) -> None """
        pass

    def __int__(self): # real signature unknown; restored from __doc__
        """ __int__(self: pyrealsense2.pyrealsense2.frame_metadata_value) -> int """
        return 0

    def __ne__(self, *args, **kwargs): # real signature unknown; restored from __doc__
        """
        __ne__(*args, **kwargs)
        Overloaded function.

        1. __ne__(self: pyrealsense2.pyrealsense2.frame_metadata_value, arg0:
        pyrealsense2.pyrealsense2.frame_metadata_value) -> bool

        2. __ne__(self: pyrealsense2.pyrealsense2.frame_metadata_value, arg0: int) -> bool
        """
        pass

    def __repr__(self): # real signature unknown; restored from __doc__
        """ __repr__(self: pyrealsense2.pyrealsense2.frame_metadata_value) -> str """
        return ""

    def __setstate__(self, arg0): # real signature unknown; restored from __doc__
        """ __setstate__(self: pyrealsense2.pyrealsense2.frame_metadata_value, arg0: tuple) -> None """
        pass

    actual_exposure = frame_metadata_value.actual_exposure 实际曝光
    actual_fps = frame_metadata_value.actual_fps 实际fps
    auto_exposure = frame_metadata_value.auto_exposure 自动曝光
    auto_white_balance_temperature = frame_metadata_value.auto_white_balance_temperature 自动白平衡温度
    backend_timestamp = frame_metadata_value.backend_timestamp 后端时间戳
    backlight_compensation = frame_metadata_value.backlight_compensation 背光补偿
    brightness = frame_metadata_value.brightness 亮度
    contrast = frame_metadata_value.contrast 对比度
    exposure_priority = frame_metadata_value.exposure_priority 曝光优先
    exposure_roi_bottom = frame_metadata_value.exposure_roi_bottom 感兴趣区域的底部曝光
    exposure_roi_left = frame_metadata_value.exposure_roi_left 感兴趣区域的左部曝光
    exposure_roi_right = frame_metadata_value.exposure_roi_right 感兴趣区域的右部曝光
    exposure_roi_top = frame_metadata_value.exposure_roi_top 感兴趣区域的顶部曝光
    frame_counter = frame_metadata_value.frame_counter 帧计数器
    frame_laser_power = frame_metadata_value.frame_laser_power 帧激光功率
    frame_laser_power_mode = frame_metadata_value.frame_laser_power_mode 帧激光功率模式
    frame_timestamp = frame_metadata_value.frame_timestamp 帧时间戳
    gain_level = frame_metadata_value.gain_level 增益水平
    gamma = frame_metadata_value.gamma 伽玛
    hue = frame_metadata_value.hue 色调
    low_light_compensation = frame_metadata_value.low_light_compensation 弱光补偿
    manual_white_balance = frame_metadata_value.manual_white_balance 手动白平衡
    power_line_frequency = frame_metadata_value.power_line_frequency 电力线频率
    saturation = frame_metadata_value.saturation 饱和
    sensor_timestamp = frame_metadata_value.sensor_timestamp 传感器时间戳
    sharpness = frame_metadata_value.sharpness 锐度
    temperature = frame_metadata_value.temperature 温度

```

Ad by **CRITEO**

Report this ad

```

    backend_timestamp : <failed to retrieve the value> ,
    'backlight_compensation': '<failed to retrieve the value>',
    'brightness': '<failed to retrieve the value>',
    'contrast': '<failed to retrieve the value>',
    'exposure_priority': '<failed to retrieve the value>',
    'exposure_roi_bottom': '<failed to retrieve the value>',
    'exposure_roi_left': '<failed to retrieve the value>',
    'exposure_roi_right': '<failed to retrieve the value>',
    'exposure_roi_top': '<failed to retrieve the value>',
    'frame_counter': '<failed to retrieve the value>',
    'frame_laser_power': '<failed to retrieve the value>',
    'frame_laser_power_mode': '<failed to retrieve the value>',
    'frame_timestamp': '<failed to retrieve the value>',
    'gain_level': '<failed to retrieve the value>',
    'gamma': '<failed to retrieve the value>',
    'hue': '<failed to retrieve the value>',
    'low_light_compensation': '<failed to retrieve the value>',
    'manual_white_balance': '<failed to retrieve the value>',
    'power_line_frequency': '<failed to retrieve the value>',
    'saturation': '<failed to retrieve the value>',
    'sensor_timestamp': '<failed to retrieve the value>',
    'sharpness': '<failed to retrieve the value>',
    'temperature': '<failed to retrieve the value>',
    'time_of_arrival': '<failed to retrieve the value>',
    'white_balance': '<failed to retrieve the value>',
}

```

problem

Unable to print actual_exposure somehow

```
print(color_frame.get_frame_metadata(rs.frame_metadata_value.actual_exposure))
```

result:

```

D:\20191031_tensorflow_yolov3\python\python.exe D:/2_pycharm测试项目/200108_测试获取Intel_Realsense_options参数/200110_测试摄像头实时获取曝光值.py
摄像头838212073161初始化成功
Traceback (most recent call last):
  File "D:/2_pycharm测试项目/200108_测试获取Intel_Realsense_options参数/200110_测试摄像头实时获取曝光值.py", line 52, in <module>
    print(color_frame.get_frame_metadata(rs.frame_metadata_value.actual_exposure))
RuntimeError: metadata not available

Process finished with exit code 1

```

In addition, auto_exposure, auto_white_balance_temperature, exposure_roi_bottom also do not work, I don't know what's going on

- [like](#)

- [collect](#)

- [share](#)

Ad by **CRITEO**

Report this ad

- - Article report



[Dontla](#)

Published 747 original articles , received 28 likes , and visited 90,000+

[his message board](#) pay attention to

you may also like

Reprinted from blog.csdn.net/Dontla/article/details/103921994

related articles



Ad by **CRITEO**

Report this ad

MELIÁ SALINAS

FREE CANCELLATION

Lanzarote

From €235

SEARCH

[Add the program to the registry and start automatically at boot](#)

[A graphical anti-kill in nim language](#)

[Application and Practice of Software Engineering in Shandong University - Using CUDA/GPU Technology to Accelerate Cryptographic Operations \(Week 10\)](#)

[Samsung Memory Coding](#)

[Samsung Memory Code The world's first 7nm EUV processor! Samsung announces Exynos 9825](#)

[Samsung memory coding Samsung Exynos 9825 officially released: the world's first 7nm EUV process mobile phone chip](#)

Meliá Salinas

Melia

Ad by **CRITEO**[Report this ad](#)

[A simple socket program - linux](#)

[Getting started with python2.7---function](#)

[The ninth week of learning summary.](#)

[Graphic and text tutorial on building and using SVN server under Windows](#)

[Jenkins creates JMeter Job for continuous integration of performance testing](#)

[java thread pool understanding](#)

[Convert xml matrix saved by OpenCV to MATLAB matrix](#)

[The sword refers to the second edition of the offer interview question 22: the last k-th node in the linked list \(java\).](#)

[The Biography of the Bad Guys: Wang Yi \(a lie\).](#)

[daily archive](#) [More](#)

[2022-01-11\(2\)](#)

[2022-01-10\(94\)](#)

[2022-01-09 \(483\)](#)

[2022-01-08 \(320\)](#)

[2022-01-07\(446\)](#)

[2022-01-06 \(349\)](#)

[2022-01-05 \(368\)](#)

[2022-01-04 \(311\)](#)

[2022-01-03 \(382\)](#)

[2022-01-02 \(312\)](#)

Code world

© 2018 [codetd.com](#)

Ad by **CRITEO**

Report this ad

