

# AI镜头模组获取uvc视频流

支持以下方法：

## 1. v4l2(Linux, 原始帧)

---

- a. 根据索引打开设备 (/dev/videoX)
- b. 调用VIDIOC\_S\_FMT设置摄像头属性 (格式以及大小)
- c. 调用VIDIOC\_REQBUFS 申请buffer
- d. 调用VIDIOC\_STREAMON打开stream
- e. 调用VIDIOC\_DQBUF获取数据帧
- f. 调用VIDIOC\_QBUF归还buffer
- g. mmap上面获取的数据帧

## 2. qvideoframe(跨平台, 支持windows, linux, 基于qt, 拿原始帧)

---

待实现

## 3. opencv 方法 (跨平台, 支持windows, linux, 非原始帧, opencv 解码后)

---

- a. 构建cv::VideoCapture对象并根据索引打开摄像头

```
cv::VideoCapture m_capture;
m_capture = cv::VideoCapture(m_camera_index);
```
- b. 设置摄像头属性 (格式与大小: mjpeg@720p)

```
m_capture.set(CV_CAP_PROP_FOURCC, CV_FOURCC('M', 'J', 'P', 'G'));
m_capture.set(CV_CAP_PROP_FPS, 30);
m_capture.set(CV_CAP_PROP_FRAME_WIDTH, 1280);
m_capture.set(CV_CAP_PROP_FRAME_HEIGHT, 720);
```
- c. 读取帧数据

```
cv::Mat frame
m_capture.read(frame);
```

## 代码示例

提供上述方法1和3的示例代码如下

C++:



camera.cpp



camera.h

Python3:

a. pip3 install opencv-python

b. python3 PyCamera-CV.py



PyCamera-CV.py