### **高帧率摄像头驱动教程**

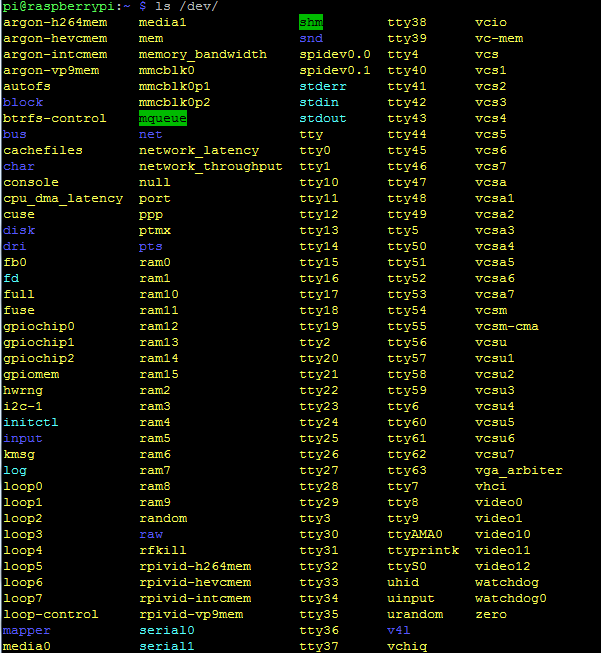
OpenCV使用到的常见API函数：

**1、cv2.VideoCapture()函数：**

cap = cv2.VideoCapture(0)

VideoCapture()中参数是0，表示树莓派video0。

（注意：可以通过命令ls /dev/ 查看当前摄像头）



cap = cv2.VideoCapture("…/1.avi")

VideoCapture("…/1.avi")，表示参数是视频文件路径则打开视频。

**2、cap.set()函数**

设置摄像头参数 不要随意修改，常见配置方法：

capture.set(CV\_CAP\_PROP\_FRAME\_WIDTH, 1920); #宽度

capture.set(CV\_CAP\_PROP\_FRAME\_HEIGHT, 1080); #高度

capture.set(CV\_CAP\_PROP\_FPS, 30); #帧数

capture.set(CV\_CAP\_PROP\_BRIGHTNESS, 1); #亮度 1

capture.set(CV\_CAP\_PROP\_CONTRAST,40); #对比度 40

capture.set(CV\_CAP\_PROP\_SATURATION, 50); #饱和度 50

capture.set(CV\_CAP\_PROP\_HUE, 50); #色调 50

capture.set(CV\_CAP\_PROP\_EXPOSURE, 50); #曝光 50

CV\_CAP\_PROP\_POS\_MSEC - 影片目前位置，为毫秒数或者视频获取时间戳

CV\_CAP\_PROP\_POS\_FRAMES - 将被下一步解压/获取的帧索引，以0为起点

CV\_CAP\_PROP\_POS\_AVI\_RATIO - 视频文件的相对位置（0 - 影片的开始，1 - 影片的结尾)

CV\_CAP\_PROP\_FRAME\_WIDTH - 视频流中的帧宽度

CV\_CAP\_PROP\_FRAME\_HEIGHT - 视频流中的帧高度

CV\_CAP\_PROP\_FPS - 帧率

CV\_CAP\_PROP\_FOURCC - 表示codec的四个字符

CV\_CAP\_PROP\_FRAME\_COUNT - 视频文件中帧的总数

函数cvGetCaptureProperty获得摄像头或者视频文件的指定属性。

**以下为详细参数：**

#define CV\_CAP\_PROP\_POS\_MSEC       0     //以毫秒计算的当前的位置

#define CV\_CAP\_PROP\_POS\_FRAMES     1  //以帧计算当前的位置

#define CV\_CAP\_PROP\_POS\_AVI\_RATIO  2  //视频的相对位置，从0 到 1 前面这三个参数应该是跟视频播放，读取相关的动态信息

#define CV\_CAP\_PROP\_FRAME\_WIDTH    3    //帧宽

#define CV\_CAP\_PROP\_FRAME\_HEIGHT   4   //帧高度

#define CV\_CAP\_PROP\_FPS            5    //帧率

#define CV\_CAP\_PROP\_FOURCC         6   //4 字符编码方式

#define CV\_CAP\_PROP\_FRAME\_COUNT    7 //视频帧数

#define CV\_CAP\_PROP\_FORMAT         8   //视频格式

#define CV\_CAP\_PROP\_MODE           9   //后端特定值，指示当前捕获模式。

#define CV\_CAP\_PROP\_BRIGHTNESS    10   //亮度

#define CV\_CAP\_PROP\_CONTRAST      11  //对比度

#define CV\_CAP\_PROP\_SATURATION    12 //饱和度

#define CV\_CAP\_PROP\_HUE           13  //色调

#define CV\_CAP\_PROP\_GAIN          14 //增益

#define CV\_CAP\_PROP\_EXPOSURE      15 //曝光

#define CV\_CAP\_PROP\_CONVERT\_RGB   16 //布尔型标记图像是否应该被转换为RGB.

#define CV\_CAP\_PROP\_WHITE\_BALANCE 17 //白平衡

#define CV\_CAP\_PROP\_RECTIFICATION 18 //立体声摄像机校正标志(注意:仅支持DC1394 v2。x端cur-rently)

**3、cap.isOpened()函数：**

返回true表示成功，false表示不成功

**4、ret,frame = cap.read()函数：**

cap.read()按帧读取视频，ret,frame是获cap.read()方法的两个返回值。其中ret是布尔值，如果读取帧是正确的则返回True，如果文件未读取到结尾，它的返回值就为False。

frame就是每一帧的图像，是个三维矩阵。

**5、cv2.waitKey()函数：**

参数是1，表示延时1ms切换到下一帧图像，参数过大如cv2.waitKey(1000)，会因为延时过久而卡顿感觉到卡顿。

参数为0，如cv2.waitKey(0)只显示当前帧图像，相当于视频暂停。

**6、cap.release()与destroyAllWindows()函数：**

cap.release()释放视频，调用destroyAllWindows()关闭所有图像窗口。

**代码实现过程**

由于我们整个教程都运行在JupyterLab中，所以必须了解里面的各种组件，这里我们需要用到图像显示组件。

1、导入库：

**import ipywidgets.widgets as widgets**

2、设置Image组件：

image\_widget = widgets.Image(format='jpeg', width=600, height=500)

* format:显示格式。
* width：宽度。
* height：高度。

3、显示Image组件：

display(image\_widget)

4、打开摄像头并读取图像：

image = cv2.VideoCapture(0) #打开摄像头

ret, frame = image.read() #读取摄像头数据

5、赋值给组件

#转化图像为jpeg并赋值给视频显示组件

image\_widget.value = bgr8\_to\_jpeg(frame)

|  |
| --- |
| import cv2  import ipywidgets.widgets as widgets  import threading  import time  #设置摄像头显示组件  image\_widget = widgets.Image(format='jpeg', width=600, height=500)  display(image\_widget) #显示摄像头组件 |
| #bgr8转jpeg格式  import enum  import cv2  def bgr8\_to\_jpeg(value, quality=75):  return bytes(cv2.imencode('.jpg', value)[1]) |
| image = cv2.VideoCapture(0) #打开摄像头  # width=1280  # height=960  # cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH,width)#设置图像宽度  # cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT,height)#设置图像高度  image.set(3,600)  image.set(4,500)  image.set(5, 30) #设置帧率  image.set(cv2.CAP\_PROP\_FOURCC, cv2.VideoWriter.fourcc('M', 'J', 'P', 'G'))  image.set(cv2.CAP\_PROP\_BRIGHTNESS, 40) #设置亮度 -64 - 64 0.0  image.set(cv2.CAP\_PROP\_CONTRAST, 50) #设置对比度 -64 - 64 2.0  image.set(cv2.CAP\_PROP\_EXPOSURE, 156) #设置曝光值 1.0 - 5000 156.0  ret, frame = image.read() #读取摄像头数据  image\_widget.value = bgr8\_to\_jpeg(frame) |
| while 1:  ret, frame = image.read()  image\_widget.value = bgr8\_to\_jpeg(frame)  time.sleep(0.010) |
| image.release() #使用完成对象记住释放掉对象，不然下一个程序使用这个对象模块会被占用，导致无法使用 |

