



# Linux SDK 安装使用说明

本手册描述了在 Linux 系统下安装相机驱动以及 SDK 的步骤，以及例子的编译和使用。

## 一、概述

s 系列和 f 系列的 USB2.0/USB3.0 工业相机可以在 Linux 上运行，包括嵌入式 Linux 和桌面系统 Ubuntu 等，本说明描述 SDK 在以下系统中的安装步骤。

Ubuntu 桌面板 20.04 LTS x64	ubuntu_x64_XX.tar.gz
Nvidia Jetson Nano	ubuntu_jetson_nano_XX.tar.gz
Raspberry Pi 4B	raspbian_pi4b_XX.tar.gz
NonoPC-T4 (rockchip RK3399)	ubuntu_rk3399_nonopc_t4_XX.tar.gz

XX 为日期。

SDK 对应架构压缩包解压后目录结构如下：

driver/ 规则文件,install.sh 时拷贝到系统文件夹,重启后不需要管理员权限打  
摄像头

JHCap2/ SDK 文件, libJHCap.a 是静态链接库, libJHCap.so 是动态链 install.sh  
时链接到了 /usr/lib。JHCap.h 是头文件。

Samples/ SDK 例子, 包含 C++, OpenCV, QT, Python 等程序使用相机的例子。

usbVideo/ 包含演示程序, 可执行程序可以查看图像和设置相机参数。

install.sh 安装脚本, 安装驱动规则和库文件。

README 安装说明。

## 二、安装准备

### 1 安装 OpenCV 库

```
>> sudo apt-get install libopencv-dev
```

### 2 通过源文件安装 OpenCV, 如使用 1 安装成功, 跳过本步骤

Tel: 0755-21505130 Fax: 0755-28136608

Website: <http://www.jinghangtech.com>



### 准备编译环境

```
>>sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev  
libswscale-dev
```

在如下网站下载 OpenCV 源代码，解压到一个录，并在同级建立一个 build 的目录。

<https://github.com/Itseez/opencv/archive/2.4.13.zip>

```
>>cd build  
>>cmake .../opencv-2.4.13  
>>make  
>>sudo make install  
>>export LD_LIBRARY_PATH=LD_LIBRARY_PATH:/usr/local/lib
```

### 3 安装 QT 及 Qtcreator

```
>>sudo apt-get install qt5-default qtcreator
```

### 4 安装 python-opencv

```
>>sudo apt-get install python-opencv
```

### 5 安装驱动和 JHCap2。

```
>>sudo sh install.sh
```

SDK 文档请参考 Windows 版本的 SDK 开发说明，重新启动后规则才能生效，以后普通用户打开摄像头不需要管理员权限(sudo)。

### 6 驱动依赖 libusb，如果没有安装或者运行报相关错误，需要安装。

```
>>sudo apt-get install libusb-1.0-0 libusb-1.0-0-dev
```

## 三、运行实例

### 1 运行基于 Qt 的演示程序 usbVideo/usbVideo。性能较低的树莓派，需要设置帧延时才能流程运行。

```
>>./usbVideo
```

### 2 编译 C++ 窗口示例 Samples/Console，采集并保存 50 张 bmp 图像。

```
>>make
```

```
>>./console
```

### 3 编译 OpenCV 示例 Samples/OpenCV，预览图像，在图像窗口按 ESC 退出。

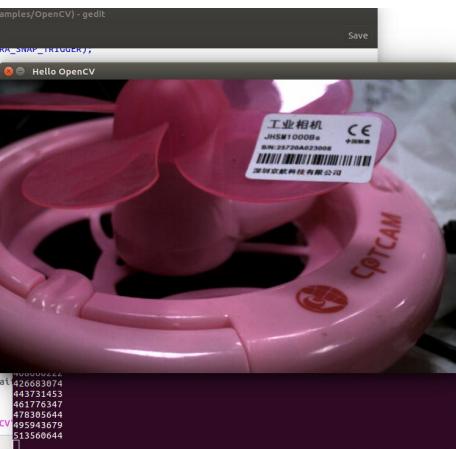
```
>>make
```

Tel: 0755-21505130 Fax: 0755-28136608

Website: <http://www.jinghangtech.com>



>>./opencv



```
openvc.cpp (-/ubuntu_x64/Samples/OpenCV)- gedit
Save
//CameraSetCapInfo(0, CVideo_Short_Kinect);
int width=370, height=40;
CameraSetResolution(0, 0, 0);
CameraGetResolution(0, 0, 0);
//CameraSetROI(0, 0, 0, width);
CameraGetImageBufferSize(0, 0);
IplImage *image = cvCreateImage(cvSize(width, height), IPL_DEPTH_8U, 3);
cvNamedWindow("Hello OpenCV");
while(true)
{
    if(CameraQueryImage(0, &image))
    {
        struct timespec t;
        t.tv_sec = 0;
        t.tv_nsec = 100000000;
        clock_gettime(CLOCK_MONOTONIC, &t);
        printf("%ld\n", t.tv_sec);
        free(t);

        //verode(image, image);
        cvShowImage("Hello OpenCV");
    }
    else
    {
        usleep(1000);
    }
    unsigned char key = cvWaitKey(426683074);
    if(key==27) break;
}
cvReleaseImage(image);
cvDestroyWindow("Hello OpenCV");
CameraFree(0);
513560644
```

4 编译 Qt 示例 Samples/QtDemo, 用 Qt Creator 打开工程编译, 本工程演示单相机采集图像和设置参数。

5 编译 Qt 示例 Samples/QtDemo\_Dual, 用 Qt Creator 打开工程编译, 本工程演示双相机采集图像和设置参数。

6 运行 Python 示例 Samples/Python, 在图像窗口按 ESC 退出.

>> python loop.py



```
loop.py (-/ubuntu_x64/Samples/Python)- gedit
Save
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)
dll.CameraSetContrast.argtypes = [ c_int, c_double ]
dll.CameraSetContrast(0, 1.15)
bufLen = c_int()
width = c_int()
height = c_int()
dll.CameraGetImage
dll.CameraGetImage
inbuf = create_string_buffer(4096)
cv2.namedWindow("ROXI 若惜")
while 1:
    dll.CameraGetImage(0, 0, width, height, inbuf, bufLen)
    arr= np.frombuffer(inbuf, dtype='uint8')
    image=arr.reshape((height.value, width.value, 3))
    cv2.imshow('ROXI 若惜', image)
    key=cv2.waitKey(1)
    if key&0xFF == 27:
        cv2.destroyAllWindows()
        break
cv2.destroyAllWindows()
```