更多嵌入式 Linux 学习资料，请关注：一口 Linux 回复关键字:1024

个人微信

公众号

一、嵌入式视频图像开源库

在嵌入式系统中，常用的视频图像处理开源系统有：luvcview、cheese、motion、mjpg-streamer 或者 ffmpeg，其中：

- luvcview: 基于 V4L2、SDL 的程序，支持拍照录像，参数调节，代码精简实用，适合学习 V4L2 编程
- cheese：基于 V4L2、GTK 的程序，支持拍照录像，特殊视频效果
- motion：移动侦测拍照程序
- mjpg-streamer：网络摄像机程序

二、mjpg-streamer 简介

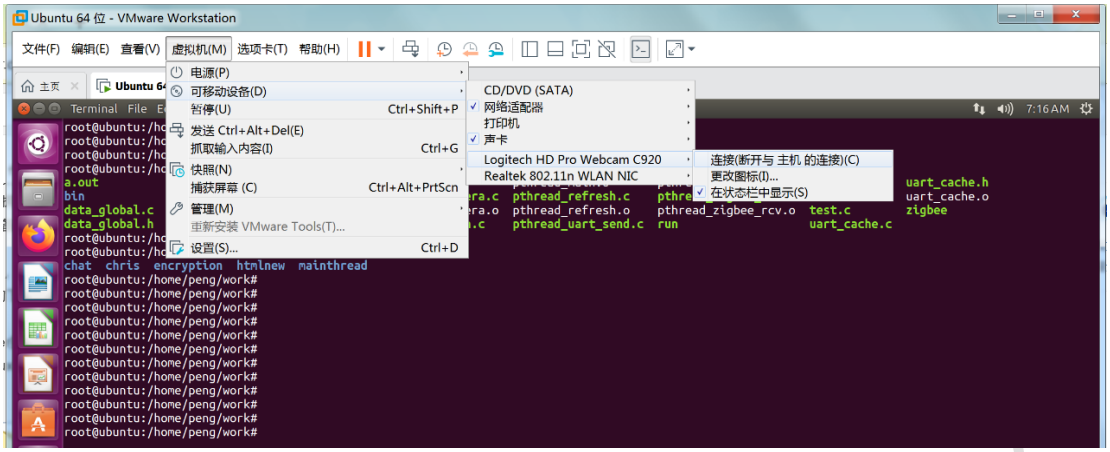
MJPG-streamer 是一个优秀的开源 project，它可以通过 HTTP 的方式访问 linux 上面的兼容摄像头，从而做到远程视频传输的效果。

MJPG-streamer 从 webcam 摄像头采集图像，把他们以流的形式通过基于 ip 的网络传输到浏览器如 Firehox，Cambozola，VLC 播放器，Windows 的移动设备或者其他拥有浏览器的移动设备。

它可以利用某些 webcams 的硬件压缩功能来降低服务器 CPU 的开销。

它为嵌入式设备和一些常规服务器提供了一个轻量且更少 CPU 消耗的方案，因为它无需为视频帧压缩浪费大量的计算效率。

三、测试摄像头



1）按上图的方式将罗技摄像头连接入虚拟机

2）下载安装 cheese 检测摄像头是否能够正常工作

```
$ sudo apt-get update
$ sudo apt-get install cheese
```

ubuntu 16.04 已经自带该程序

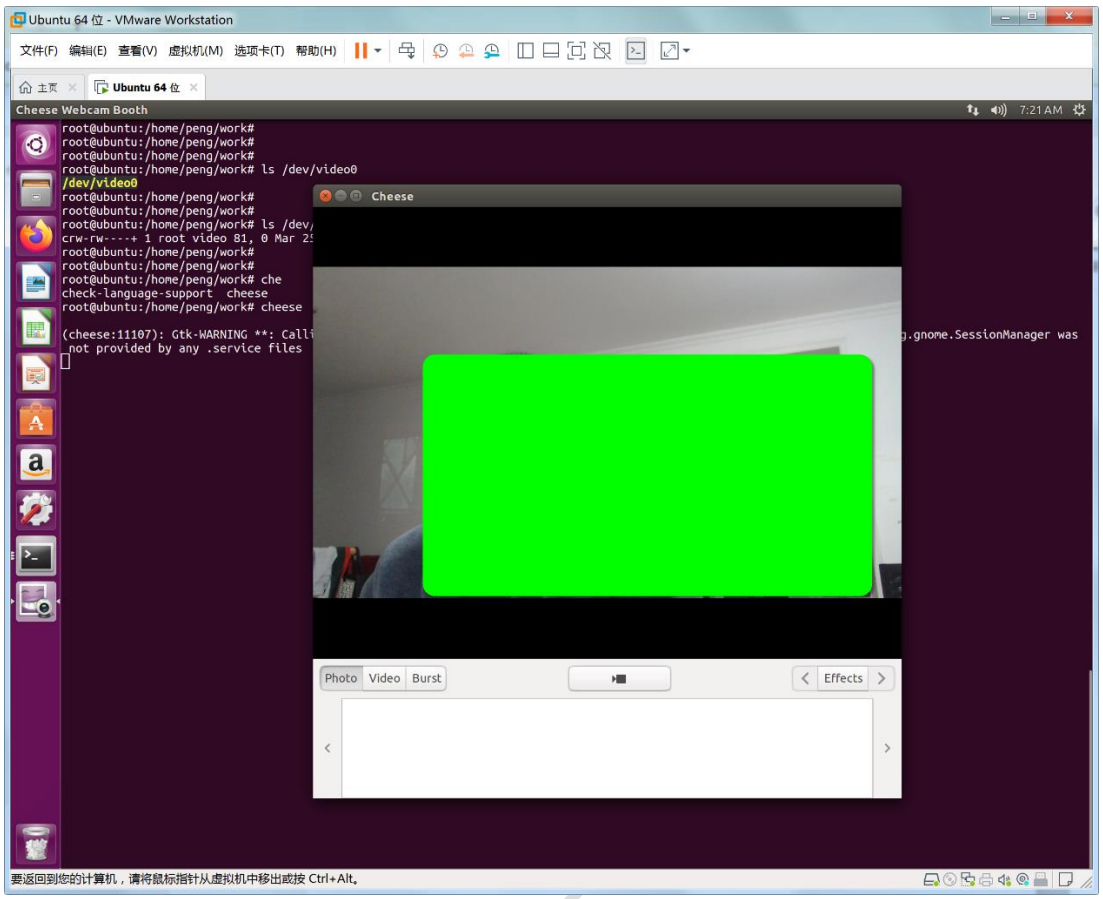
3）测试

摄像头连接后会生成以下设备文件

```
root@ubuntu:/home/peng/work# ls /dev/video0 -l
crw-rw----+ 1 root video 81, 0 Mar 25 07:18 /dev/video0
```

运行

```
root@ubuntu:/home/peng/work# cheese
```



四、移植

安装准备:

1. 安装前准备

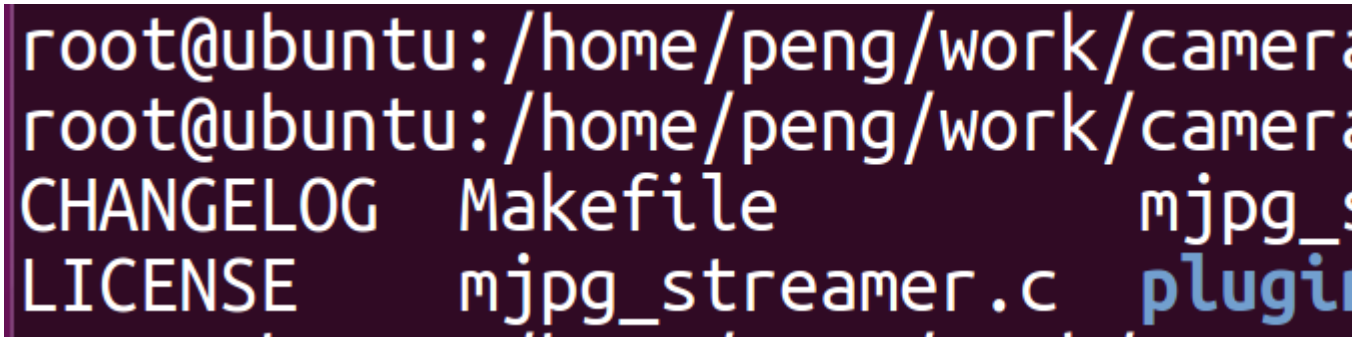
sudo apt-get install libsdl1.2-dev subversion
sudo apt-get install libjpeg62-dev
sudo apt-get install imagemagick

2. 下载 mjpeg-streamer

git clone https://github.com/shrkey/mjpg-streamer
如果没有安装 git,执行以下命令
sudo apt-get install git

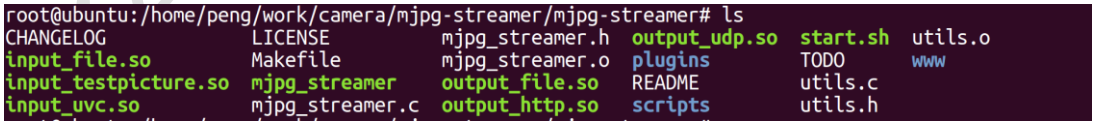
3. 编译安装

cd mjpg-streamer/mjpg-streamer



root@ubuntu:/home/peng/work/camera/mjpg-streamer# tree -L 1 ./
./
├── doc
├── mjpeg-client　　#分别有 linux 和 windows 的客户端
├── mjpg-streamer　　#目录下提供了 的执行程序和各个输入输出设备组件
├── mjpg-streamer-experimental
├── mjpg-streamer.tar.gz
├── README.md
├── udp_client
└── uvc-streamer　　#目录下提供了 uvc-streamer 的可执行目录

6 directories, 2 files
ps：重新编译前，需要执行
make
sudo make install
root@ubuntu:/home/peng/work/camera/mjpg-streamer/mjpg-streamer# make install
install --mode=755 mjpg_streamer /usr/local/bin
install --mode=644 input_uvc.so output_file.so output_udp.so output_http.so input_testpicture.so input_file.so /usr/local/lib/
install --mode=755 -d /usr/local/www
install --mode=644 -D www/* /usr/local/www



编译生成的库文件功能

(1)input_testpicture.so。这是一个图像测试插件，它将预设好的图像编译成一个头文件，可以在没有摄像头的情况下传输图像，从而方便调试程序。

(2)input_uvc.so。此文件调用 USB 摄像头驱动程序 V4L2，从摄像头读取视频数据。

(3)input_control.so。这个文件实现对摄像头转动的控制接口。

(4)output_http.so。这是一个功能齐全的网站服务器，它不仅可以从单一文件夹中处理文件，还可以执行一定的命令，它可以从输入插件中处理一幅图像，也可以将输入插件的视频文件根据现有 M-JPEG 标准以 HTTP 视频数据服务流形式输出。

(5)output_file.so。这个插件的功能是将输入插件的 JPEG 图像存储到特定的文件夹下，它可以用来抓取图像。

4 修改脚本

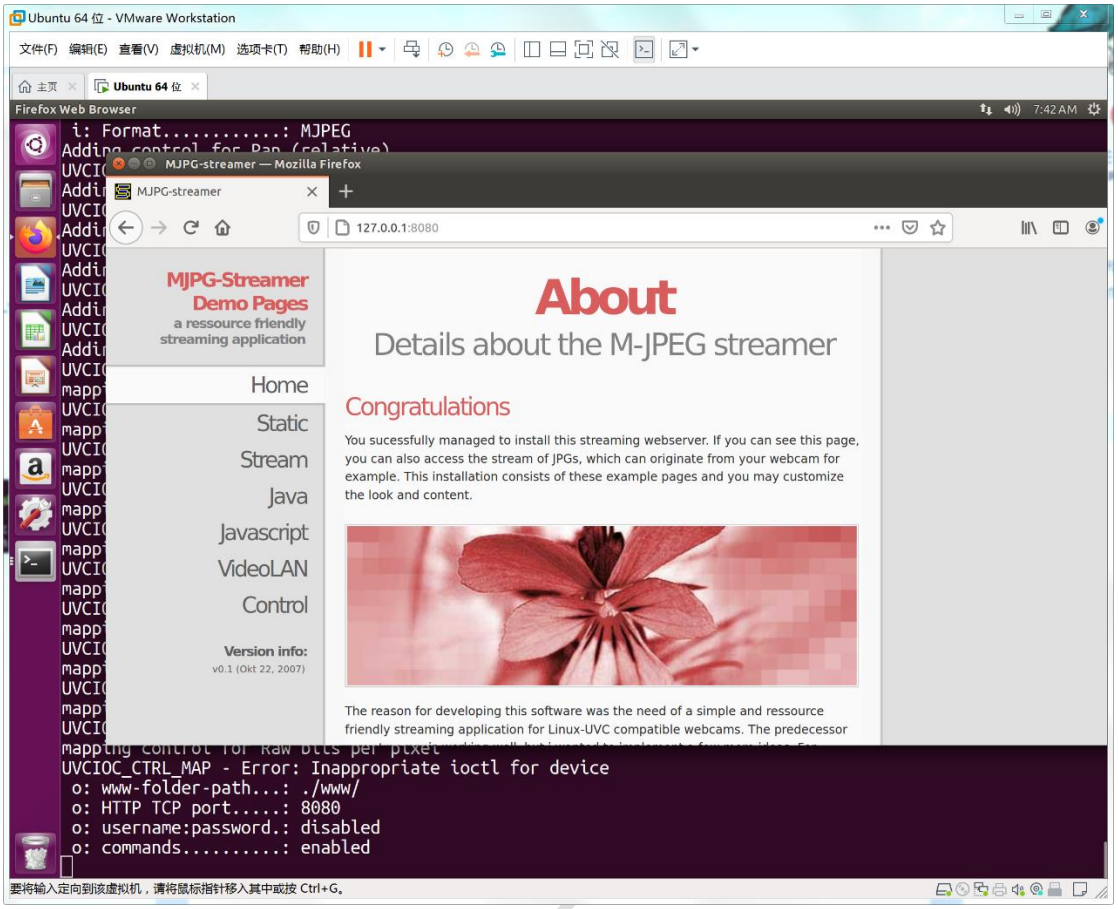
修改脚本文件
/home/peng/work/camera/mjpg-streamer/mjpg-streamer/start.sh
./mjpg_streamer -i "./input_uvc.so -y" -o "./output_http.so -w ./www" -o "./output_file.so -f /www/pice -d 15000"

```
25 ## This example shows how to invoke mjpg-streamer from the command line
26
27 export LD_LIBRARY_PATH="$(pwd)"
28 #./mjpg_streamer -i "input_uvc.so --help"
29 ./mjpg_streamer -i "./input_uvc.so -y" -o "./output_http.so -w ./www" -
30
31 #./mjpg_streamer -i "./input_uvc.so" -o "./output_http.so -w ./www"
```

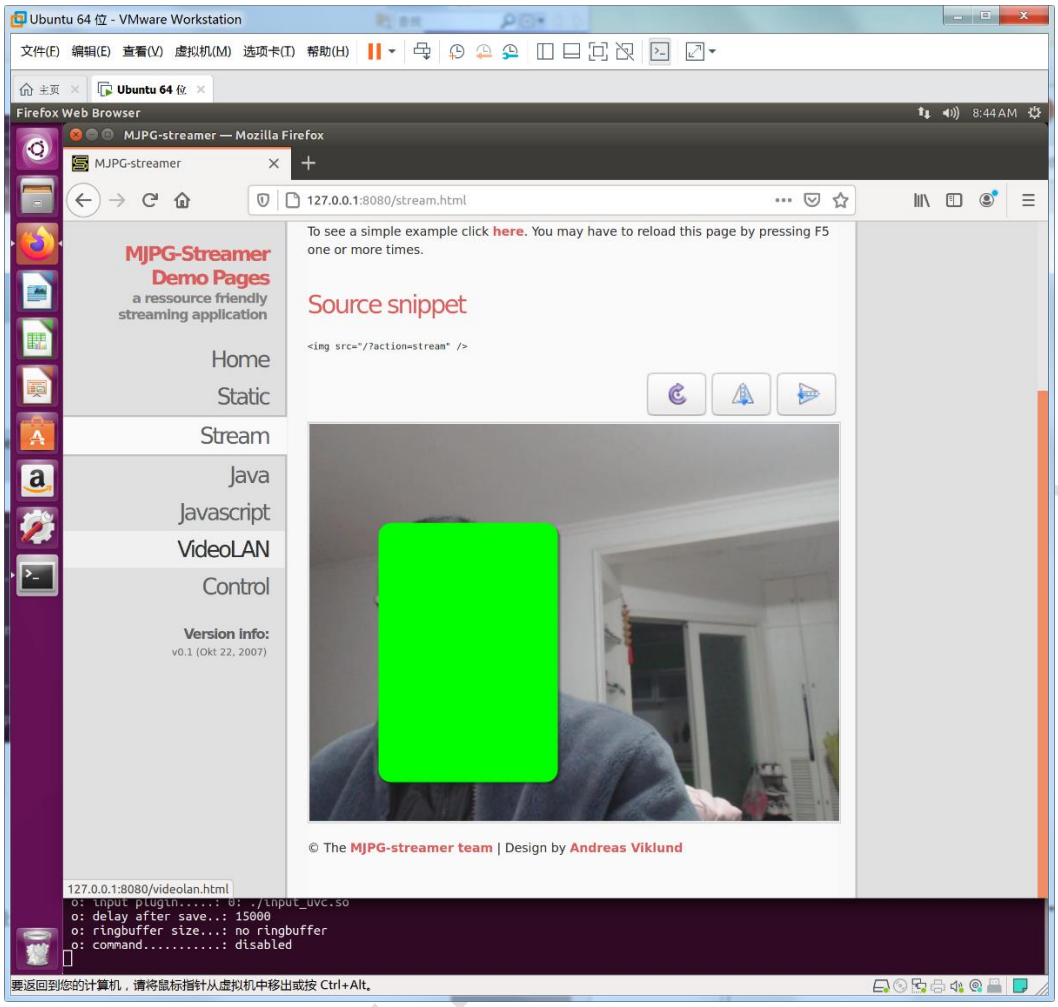
"./input_uvc.so -y" ： 指定摄像头是 YUV，默认是 JPEG，一口君使用的罗技摄像头是 YUV
"./output_http.so -w ./www" ： 指定 web 服务器根目录./www，我们可以通过浏览器测试摄像头
"./output_file.so -f /www/pice -d 15000"：指定拍照保存照片目录/www/pice，并且每 15s 保存一次照片
也可以指定分辨率:
./mjpg_streamer -i "input_uvc.so -d /dev/video0 -n -y -r 640x480 -f 30" -o "./output_http.so -w ./www" -o "./output_file.so -f /www/pice -d 15000"
市面上有的摄像头支持格式有 YUV，MJPEG，H264 ；
mjpg-stream 支持 MJPEG 和 YUV 两种格式

5 测试

运行
./start.sh
（1）网页测试


Ubuntu 64 位 - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
Firefox Web Browser
7:42 AM
i: Format.....: MJPEG
MJPG-streamer — Mozilla Firefox
MJPG-streamer
127.0.0.1:8080
MJPG-Streamer
Demo Pages
a ressource friendly
streaming application
Home
Static
Stream
Java
Javascript
VideoLAN
Control
Version info:
v0.1 (Okt 22, 2007)
About
Details about the M-JPEG streamer
Congratulations
You sucessfully managed to install this streaming webserver. If you can see this page, you can also access the stream of JPGs, which can originate from your webcam for example. This installation consists of these example pages and you may customize the look and content.
The reason for developing this software was the need of a simple and ressource friendly streaming application for Linux-UVC compatible webcams. The predecessor
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
o: www-folder-path...: ./www/
o: HTTP TCP port.....: 8080
o: username:password.: disabled
o: commands.....: enabled
要将输入定向到该虚拟机，请将鼠标指针移入其中或按 Ctrl+G。

（2）网页视频流测试



(3) 拍照功能实现

浏览器上地址栏输入如下内容:

http://127.0.0.1:8080/?action=snapshot

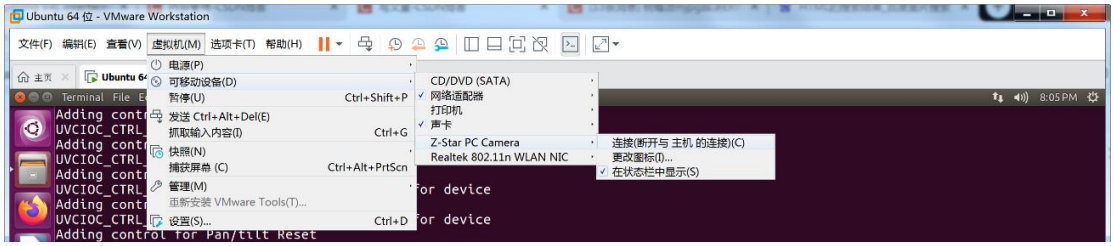
或者

http://127.0.0.1:8080/?action=stream

snapshot 表示每次抓拍一张图形显示在网页上，stream 表示视频流也就是连续的图像

6. 补充

一口君还使用了一款 z-star 摄像头，该款摄像头不要添加-y 选项



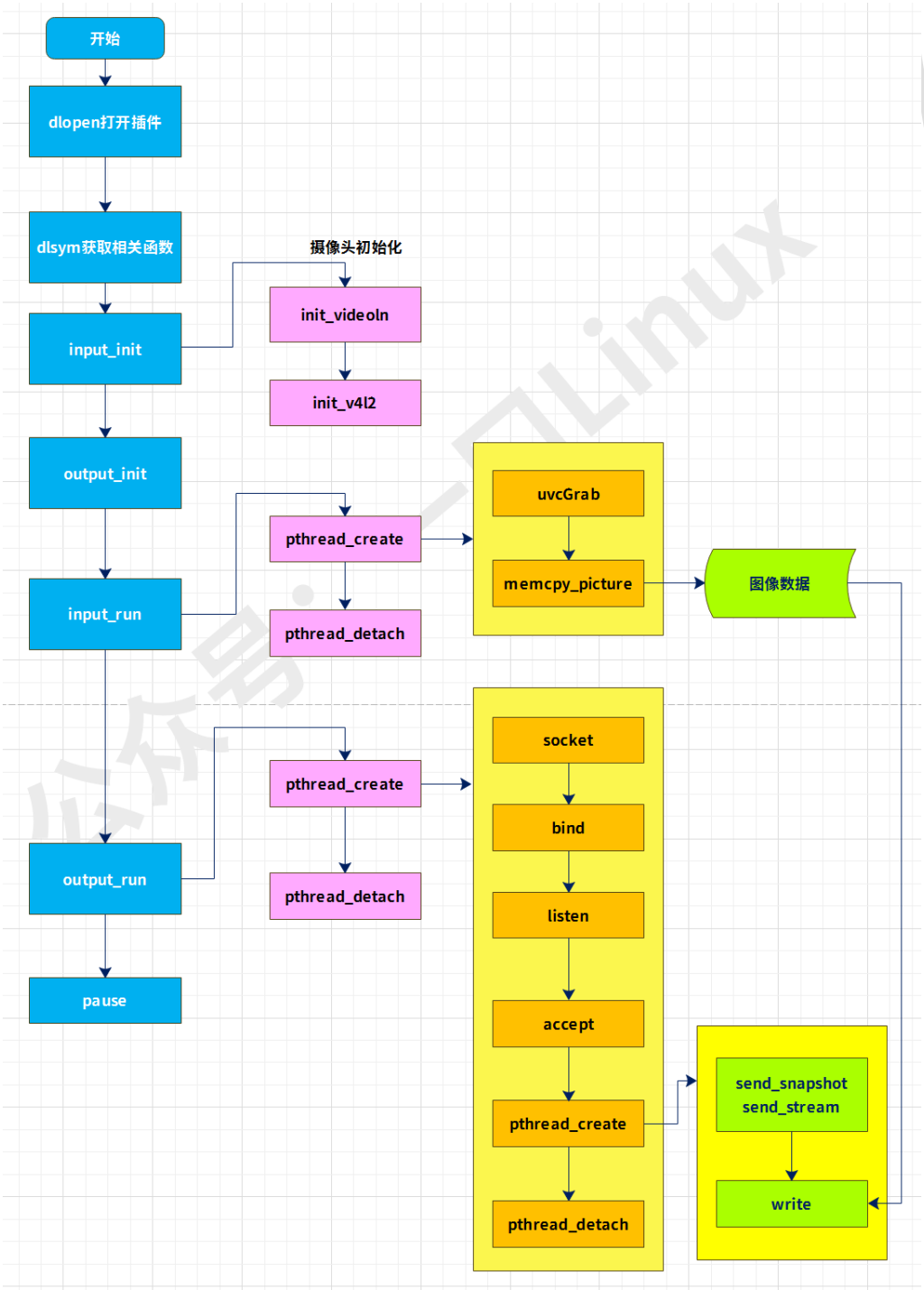
./mjpg_streamer -i "./input_uvc.so -d /dev/video0" -o "./output_http.so -w ./www" -o "./output_file.so -f /www/pice -d 150000"

有时候摄像头生成的设备文件不是/dev/video0

```
root@ubuntu:/home/peng/work/camera/mjpg-streamer/mjpg-streamer# ls /dev/video1 -l
crw-rw----+ 1 root video 81, 1 Mar 25 19:55 /dev/video1
```

我们需要对应参数:
-d /dev/video1

五、代码流程



六、支持单拍、连拍

由于 mjpg_stream 中 output-file.so 能实现连续拍照的功能，不能实现单拍或 连拍几张的功能所以需要对 output_file 原码进行修改。

peng@ubuntu:~/work/camera/mjpg-streamer/mjpg-streamer/plugins/output_file/output_file.c

1. 在 196 行 函数 voidworker_thread(voidarg) 体中加入以下代码:

```
char buf[10];    //用于存放从管道读取的命令
int flags = 0;    //拍照标志，1：表示 11 张照片，2：表示 1 张照片
int fd_com = 0; //打开管道的文件描述符

int stop_num = 0; //拍照计数

if ( access("/tmp/webcom",F_OK) < 0 )      //创建有名管道用于接收拍照命令
{
    if ( mkfifo("/tmp/webcom",0666 ) < 0)
    {
        printf("photo fifo create failed\n");
    }
}

fd_com = open ("/tmp/webcom",O_RDONLY,0666);
if (fd_com < 0)
{
    perror ("open the file webcom error");
}
```

2. 在 229 行 while(ok >= 0 && !pglobal->stop){ 后加入

```
if (flags == 0)
    {
        while(1)
        {
            read(fd_com,buf,sizeof(buf));
            if (strncmp(buf,"danger",6) == 0)      //拍 11 张照片
            {
                flags = 1;
                bzero(buf,sizeof(buf));
                break;
            }
```

```
                if (strncmp(buf,"one",3) == 0)     //拍 1 张照片
                {
                        flags = 2;
                        bzero(buf,sizeof(buf));
                        break;
                }
        }
    }
```

3. 在 355 行

```
355             /* if specified, wait now */
356             if(delay > 0) {
357                     usleep(1000 * delay);
358             }
```

后加入

```
        stop_num++;
        if (flags == 1)             //判断拍照的数量
        {
                if   ( stop_num > 9)
                {
                        stop_num = 0;
                        flags = 0;
                }
        }
        else if (flags == 2)
        {
                stop_num = 0;
                flags = 0;
        }
```