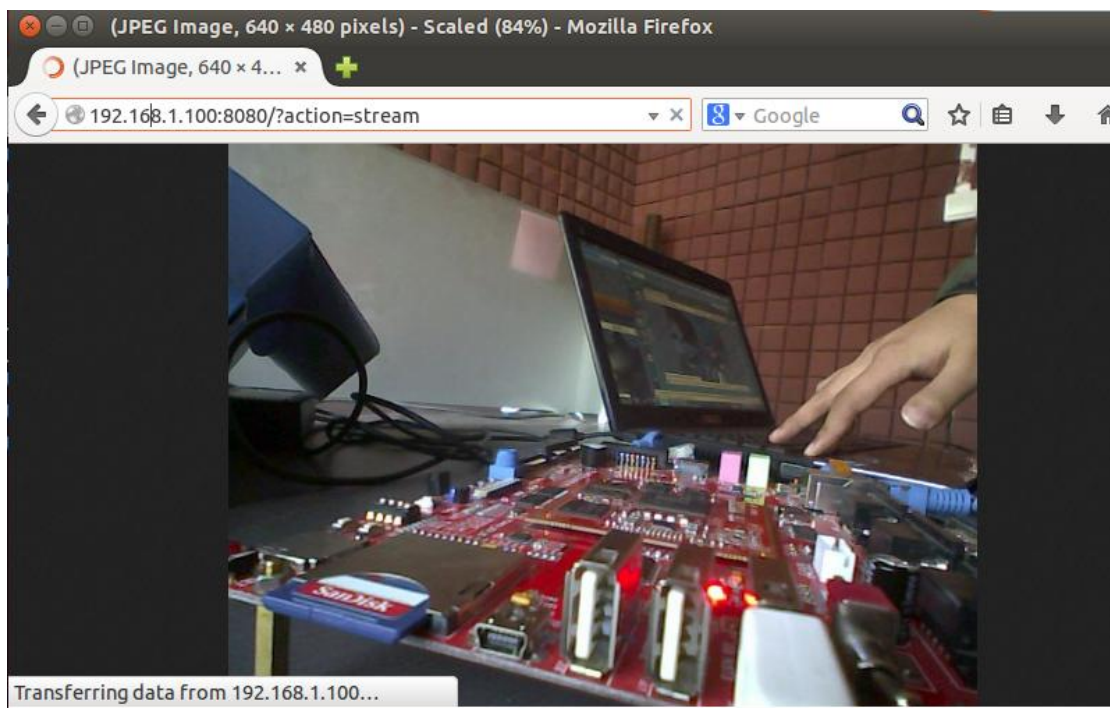


十、mjpeg-streamer 视频图像的显示



1、MJPG 简介

MJPG 是 MJPEG 的缩写,但是 MJPEG 还可以表示文件格式扩展名.

MJPEG 全名为 “Motion Joint Photographic Experts Group”, 是一种视频编码格式,

Motion JPEG 技术常用与闭合电路的电视摄像机的模拟视频信号“翻译”成视频流,并存储在硬盘上。典型的应用如数字视频记录器等。MJPEG 不像 MPEG,不使用帧间编码,因此用一个非线性编辑器就很容易编辑。

MJPEG 的压缩算法与 MPEG 一脉相承,功能很强大,能发送高质图片,生成完全动画视频等。

但相应地,MJPEG 对带宽的要求也很高,相当于 T-1,MJPEG 信息是存储在数字媒体中的庞然大物,需要大量的存储空间以满足如今多数用户的需求。

因此从另一个角度说,在某些条件下,MJPEG 也许是效率最低的编码/解码器之一。

MJPEG 是 24-bit 的 “true-color” 影像标准,MJPEG 的工作是将 RGB 格式的影像转换成 YCrCb 格式,目的是为了减少档案大小,

一般约可减少 $1/3 \sim 1/2$ 左右。

MJPEG 与 MJPG 的区别：

- 1、mjpeg 是视频，就是由系列 jpg 图片组成的视频。
- 2、MJPG 是 MJPEG 的缩写，但是 MJPEG 还可以表示文件格式扩展名。

2、环境

主机环境：ubuntu 12.04.4 LTS
目标机：Cortex-A9
主机工具链：gcc-4.6.4
交叉工具链：arm-none-linux-gnueabi-
摄像头：xxx

3、查看我的摄像头信息

14. 1 确定 USB 摄像头支持 UVC （在 PC 上）

那什么 USB 摄像头适合我们这一章的教程呢，这里有几个关键字： 1.支持 UVC（免驱）， 2.YUV 或者 MJPEG 格式输出。

14.1.1 把摄像头插入 PC 机的 USB 接口，查看 ID

注：如果你是在 Ubuntu 等 linux 操作系统下请看 1~2，在 windows 下请直接看看 3 。1. 在 linux 类操作系统下插入 USB 摄像头，用 dmesg 打印信息

```
#dmesg
```

```
[127598.091966] usb 1-1: new high-speed USB device number 29 using ehci-pci
[127598.519721] usb 1-1: New USB device found, idVendor=057e, idProduct=030a
[127598.519728] usb 1-1: New USB device strings: Mfr=48, Product=96,
SerialNumber=0
[127598.519731] usb 1-1: Product: USB Camera
[127598.519733] usb 1-1: Manufacturer: Guillemot Corporation
[127598.560887] uvcvideo: Found UVC 1.00 device USB Camera (057e:030a)
[127598.574918] input: USB Camera as
/devices/pci0000:00/0000:00:11.0/0000:02:03.0/usb1/1-1/1-1:1.0/input/input10
```

Found UVC 1.00 device USB Camera 也就是我们插入的 USB 摄像头他的 VID:PID 是 057e:030a。

这里的 ID 号可以在下一步 UVC 官方的文档中进一步确定是否被支持。

如果将用 ls /dev/video* 查看设备节点

```
fengjunhui@ubuntu:~$ ls /dev/video0
/dev/video0
```

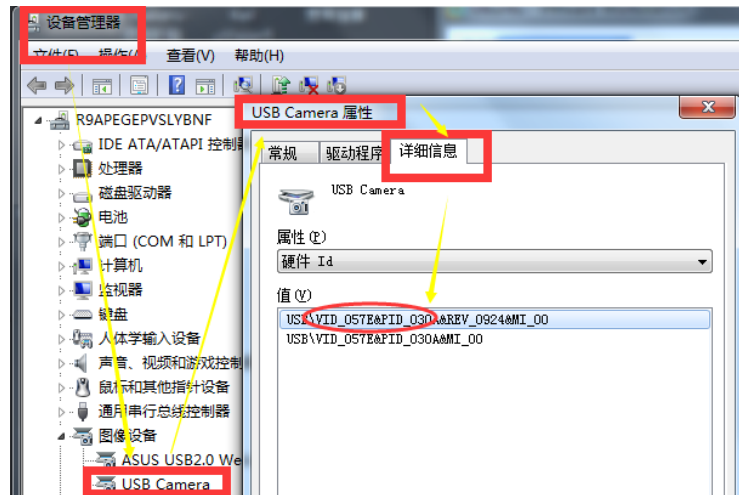
这里的 video0 是我们刚接入的 USB 摄像头。

3. 在 windows 操作系统下插入 USB 摄像头插入，打开设备管理器



第一个 USB2.0 Camera 是我们接入的 USB 摄像头，第二个 USB2.0 UVCPC Camera 是笔记本自带的摄像头。

右键属性 -> 详细信息 -> 属性 选择硬件 ID 查看



可以得到插入的 USB 摄像头 VID:PID 为 057e: 030a。这里的 ID 号可以在下一步 UVC 官方的文档中进一步确定是否被支持。

14. 1.2 确定 USB 摄像头种类

通过这个文档《摄像头驱动 VID+PID 大全》来确定芯片类型，这个文件在附带的文件夹下；通过这个网页 <http://www.ideasonboard.org/uvic/> 来查看是否支持 UVC，这个网站是 USB Video Class Linux device driver 的主页，里面有 UVC 的详细的介绍。根据前面的打印信息，根据自己的 ID 号，WEBEE 这里是搜索 USB 摄像头的 VID 号：1e4e 和 PID 号：0102。

通过摄像头的 ID，可以看到该摄像头是否支持 UVC 和其他信息。绿勾代表支持，成功的看到，我们的摄像头不支持，哈哈。

04f2:b230	HP HD Webcam [Fixed] (HP ??? notebooks)	Chicony Electronics	✓ [3]
058f:3820	Future Boy PC USB Webcam (Alcor Micro AU3820 chipset)	Alcor Micro	✓
058f:a001	HP Webcam-101 (HP Pavilion g6 notebooks)	Alcor Micro	✓
05a9:2640	OmniVision OV2640 (Dell Inspiron 1420/1720 notebooks)	OmniVision	✓
05a9:2643	OmniVision Monitor Webcam (Dell SP2208WFP)	OmniVision	✓
05a9:2649	OmniVision Monitor Webcam (Dell SP2309W)	OmniVision	✓
05a9:264b	Dell Studio Hybrid 140g	OmniVision	✓
05a9:7670	OmniVision OV7670 (Dell XPS m1330 notebooks)	OmniVision	✓
05ac:8502	Apple built-in iSight	Apple	✓ [4]
05c8:0103	FO13FF-65 PC-CAM	Foxlink	✓
05c8:0403	HP Webcam [2 MP Fixed] (HP Mini 5103 netbook)	Foxlink	✓
05ca:181c	Laptop Integrated Webcam FHD (Dell Latitude E6520 notebooks)	Ricoh	✓
05ca:18a1	Integrated Webcam (Dell Studio 1535 notebooks)	Ricoh	✓
05ca:18b7	Sony Visual Communication Camera (Sony VPCS12J1F notebooks)	Ricoh	✓

14.1.3 安装并使用 xawtv 测试 (Ubuntu 下)

```
1. 安装 xawtv 测试软件
#sudo apt-get install xawtv
2. 执行 xawtv 后面带 usb 摄像头的设备节点
#xawtv /dev/video0

或者：（推荐）
2) 使用应用程序茄子 (cheese)
    输入命令: sudo apt-get install cheese
    装好后，终端输入命令:
cheese
，即可打开。
```

4、驱动移植过程

1，添加驱动支持 USB 储存设备。

1. 进入 USB support (需要让 USB 设备正常工作，串口设备正常工作)

添加驱动支持 USB 储存设备。

(根据《系统移植实验手册-V2.pdf》[在我们的设备树中添加 USB 的设备信息](#)) ==== 关键步骤

```
进入内核目录: cd linux-3.14
#make menuconfig
```

```
Device Drivers --->
[*] USB support --->
    <*> EHCI HCD (USB 2.0) support
    <*> EHCI support for Samsung S5P/EXYNOS SoC Series
    <*> USB Mass Storage support
    <*> USB3503 HSIC to USB20 Driver
    USB Physical Layer drivers --->
        <*> Samsung USB 2.0 PHY controller Driver
```

```
SCSI device support --->
  <*> SCSI device support
  <*> SCSI disk support
```

2. 选中 Multimedia support

Device Drivers --->

<*> Multimedia support --->

如图配置:

```
-- Multimedia support
*** Multimedia core support ***
[*] Cameras/video grabbers support
[ ] Analog TV support
[ ] Digital TV support
[ ] AM/FM radio receivers/transmitters support
[ ] Remote Controller support
[*] Media Controller API (EXPERIMENTAL)
[*] V4L2 sub-device userspace API (EXPERIMENTAL)
[ ] Enable advanced debug functionality on V4L2 drivers
[ ] Enable old-style fixed minor ranges on drivers/video devices
*** Media drivers ***
[*] Media USB Adapters --->
[*] V4L platform devices --->
[ ] Memory-to-memory multimedia devices --->
[ ] Media test drivers --->
*** Supported MMC/SDIO adapters ***
[ ] Autoselect tuners and i2c modules to build
*** Media ancillary drivers (tuners, sensors, i2c, frontends) ***
Encoders, decoders, sensors and other helper chips --->
Sensors used on soc_camera driver --->
Customise DVB Frontends --->
```

再进入 Media USB Adapters

Device Drivers --->

<*> Multimedia support --->

<*>Media USB Adapters --->

如图配置

```
-- Media USB Adapters
*** Webcam devices ***
<*> USB Video Class (UVC)
[*]   UVC input events device support
<*> GSPCA based webcams --->
< > USB Philips Cameras
<*> CPiA2 Video For Linux
< > USB ZR364XX Camera support
```

4. 进入 GSPCA base webcams

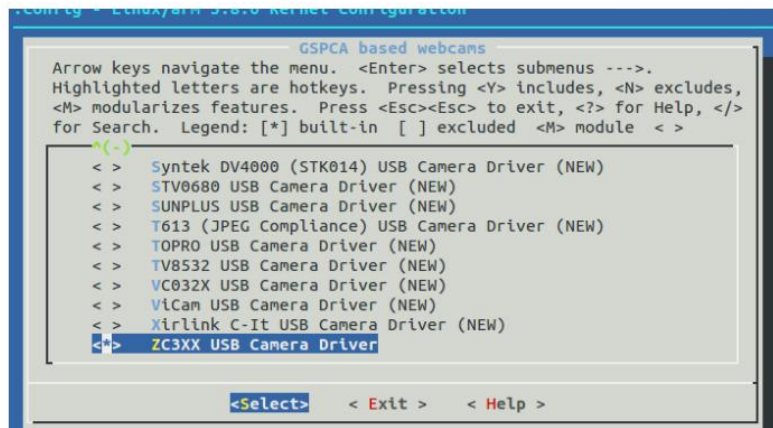
Device Drivers --->

<*> Multimedia support --->

<*>Media USB Adapters --->

<*>GSPCA base webcams

如图配置



5. 进入 V4L platform devices

Device Drivers --->

<*> Multimedia support --->

<*>V4L platform devices --->

如图配置

```
--- V4L platform devices
<*> Support for timberdale Video In/LogiWIN
<*> SoC camera support
<*> platform camera support
< > R-Car Video Input (VIN) support
< > SuperH Mobile MIPI CSI-2 Interface driver
< > SuperH Mobile CEU Interface driver
[ ] Samsung S5P/EXYNOS4 SoC series Camera Subsystem d
```

6. 编译内核

#make uImage

重新拷贝到我们的 tftp 目录下。

5、mjpg-stream 的移植

关于 mjpg-stream 的资料大家可以在下面这个网址查看：

http://sourceforge.net/apps/mediawiki/mjpg-streamer/index.php?title=Main_Page

```
source code 的网址是: http://mjpeg-streamer.svn.sourceforge.net/viewvc/mjpeg-streamer/
```

mjpeg-stream 的移植需要 jpeg 的库，所以我们先移植 jpeg 的库

(1)jpeg 库的移植

1)jpeg 源码包通过下面这个网址下载

```
http://www.ijg.org/files/jpegsrc.v8b.tar.gz
```

2)解压源码包

```
tar xvf jpegsrc.v8b.tar.gz
```

3)配置源码

```
cd jpeg-8b
./configure --prefix=/home/fengjunhui/securitymonitor/mjpeg-streamer/video/jpegarm --host=arm-none-linux-gnueabi
```

(注意：--prefix 表示最终生成的库和头文件等存放的目录，自己指定一个存在的路径就可以，

后面编译 mjpeg-streamer 的时候回用到，此处必须使用绝对路径，--host=arm-none-linux-gnueabi --host 表示生成的库的运行平台，注意是两个 -，末尾不能加-，./configure 之后为一行命令)

```
比如我的路径: pwd
/home/fengjunhui/securitymonitor/mjpeg-streamer/video/jpegarm
(这个目录建议自己创建一个在 mjpeg-streamer 的上一级目录下)
```

4)编译

```
make
```

5)安装

```
make install
```

6)拷贝库到文件系统中

```
sudo cp /home/fengjunhui/securitymonitor/mjpeg-streamer/video/jpegarm/lib/* /home/fengjunhui/source/rootfs/lib
```

(2)、移植 mjpeg-streamer

1、下载源码，在 <https://sourceforge.net/projects/mjpeg-streamer/> 下载的源码(mjpeg-streamer-code-182) 最新的版本使用以下命令下载，

```
svn cohttps://mjpg-streamer.svn.sourceforge.NET/svnroot/mjpg-streamer/mjpg-streamer
```

解压，并进入目录一共有 这么多目录

```
doc mjpg-streamer udp_client
mjpeg-client mjpg-streamer-experimental uvc-streamer
mjpg-streamer : 目录下提供了 的执行程序和各个输入输出设备组件
uvc-streamer   : 目录下提供了 uvc-streamer 的可执行目录
mjpeg-client:  : 分别有 linux 和 windows 的客户端
```

2、进入其目录

```
cd /home/mjpg/mjpg-streamer-code-182/mjpg-streamer
```

3、修改源码

修改

```
vi ./plugins/input_uvc/Makfile,
```

修改

```
CFLAGS += -O1 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC
```

为

```
CFLAGS += -O1 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -I
/home/fengjunhui/securitymonitor/mjpeg-streamer/video/jpegarm/include
```

修改

```
(CC)(CFLAGS) -o @inputuvc.cv4l2uvc.lojpegutils.lodyncntrl.lo(LFLAGS)
```

为（注意还是一行，不要直接复制粘贴，根据实际情况思考来做）

```
(CC)(CFLAGS) -L /home/fengjunhui/securitymonitor/mjpeg-streamer/video/jpegarm/lib -o
@inputuvc.cv4l2uvc.lojpegutils.lodyncntrl.lo(LFLAGS)
```

注： /home/fengjunhui/securitymonitor/mjpeg-streamer/video/jpegarm/lib 就是上面移植 jpeg 库后生成的 lib、include 的目录的路径，千万不要写错了

4、在 mjpg-streamer-code-182/mjpg-streamer 目录下

```
编译: make CC=arm-linux-gcc
在 mjpg-streamer 生成了
input*.so output*.so 和 mjpg_streamr 可执行文件
小问题:
make CC=arm-linux-gcc
make: svnversion: Command not found
解决方案: sudo apt-get install subversion
安装之前先更新一下软件包的源和依赖:
```



```
Sudo apt-get update          更新软件源
Sudo apt-get -f install      更新依赖
```

再次编译:

```
make
```

```
Nothing to be done for `all'.
```

先

```
make clean,
```

然后再

```
make CC=arm-linux-gcc
```

查看生成的库和可执行的文件:

```
fengjunhui@ubuntu:~/securitymonitor/mjpeg-streamer/mjpg-streamer-code-182/mjpg-streamer$ ls *.so
input_file.so          input_uvc.so          output_http.so
input_testpicture.so   output_file.so        output_udp.so
ls mjpg_streamer
mjpg_streamer
```

5、在开发板建立 mjpg-streamer 安装目录 (关键性的拷贝过程)

```
mkdir    /home/fengjunhui/source/rootfs/mjpg-streamer (你的根文件系统目录)
cp *.so   /home/fengjunhui/source/rootfs/mjpg-streamer -arf
cp mjpg_streamer /home/fengjunhui/source/rootfs/mjpg-streamer -arf
```

将源码目录中的 start.sh 和目录 www 拷贝到~/source/rootfs/mjpg-streamer/目录下, 然后就可以测试啦

```
sudo cp start.sh www/ ~/source/rootfs/mjpg-streamer/ -arf
```

6、测试:

```
./start.sh (记得插上摄像头设备喔)
```

MJPEG 格式错误,

```
ERROR opening V4L interface: No such file or directory
Init v4L2 failed !! exit fatal
i: init_VideoIn failed
MJPG-streamer [1296]: init_VideoIn failed
```

解决方案:

1、插上 USB 摄像头, 查看设备节点: `ls /dev/video0`

2、如果没有设备节点, 则参考 [《ttyUSB0 设备节点不显示的解决方案.doc》](#)

视频流格式错误:

```
cannot be displayed because it contains errors
修改 start.sh 文件, 加参数-y, 然后运行 start.sh;
./mjpg_streamer -i “./input_uvc.so -y” -o “./output_http.so -
w ./www”
```

在 PC 机打开一个网页，输入 `http://192.168.1.100:8080/?action=stream` 就可以看到图像。
(板子 ip)

以上是 mjpg-streamer 的移植全过程，下面说说在这过程可能出现的错误以及解决方法。

1、当编译 mjpg-stream-r63 时出错

```
[root@wu mjpg-stream-r63]# make CC=arm-linux-gcc
arm-linux-gcc -O3 -DLINUX -D_GNU_SOURCE -Wall -c -o mjpg_streamer.o
mjpg_streamer.c
mjpg_streamer.c:27:28: fatal error: linux/videodev.h: No such file or
directory
compilation terminated.
make: * [mjpg_streamer.o] 错误 1
```

原因：这是因为内核 2.6.38 和以后的版本已经移除了对 v4l1 的支持，这个东西可能比较老旧了。新的头文件是 `video4linux2.h`；相比 v4l1 很多接口都变化。

解决办法：可以选择以前低版本内核，或者用新版本的 `mjpg-streamer-code-182`，下载地址在上面有链接。

3、使用摄像头出现如下信息

```
.....
Init v4L2 failed !! exit fatal
init_VideoIn failed
.....
```

原因：市面上大部分摄像头都是支持 YUV 的，而不是 JPEG 的。

解决方法：由于 mjpg-stream 支持 JPEG 和 YUV 两种格式，所以只需修改 `start.sh`，在输入中加入 “ -y ”

```
假如原来是 ./mjpg_streamer -i “./input_uvc.so ” -o “./output_http.so -
w ./www”，则改为
./mjpg_streamer -i “./input_uvc.so -y” -o “./output_http.so -
w ./www” 即可。
```

总结：经过一轮的谷歌+百度后终于可以在 firefox 看了摄像头采集的图像

4、缺少库的支持

```
root@makeru :/mjpg-streamer# ./start.sh
MJPEG-streamer [1308]: starting application
MJPEG Streamer Version: svn rev:
MJPEG-streamer [1308]: MJPEG Streamer Version: svn rev:

ERROR: could not find input plugin
MJPEG-streamer [1308]: ERROR: could not find input plugin
```

```

    Perhaps you want to adjust the search path with:
MJPEG-streamer [1308]:      Perhaps you want to adjust the search path with:

    # export LD_LIBRARY_PATH=/path/to/plugin/folder
MJPEG-streamer [1308]:      # export LD_LIBRARY_PATH=/path/to/plugin/folder

    dlopen: libjpeg.so.8: cannot open shared object file: No such file or directory
MJPEG-
streamer [1308]:      dlopen: libjpeg.so.8: cannot open shared object file:
No such file or directory

```

忘记拷贝库（这里需要保证你拷贝的库是 ARM 格式的 file libjpeg.so.8 ）

```

cp /home/fengjunhui/securitymonitor/mjpeg-streamer/video/jpegarm/lib/
/home/fengjunhui/source/rootfs/lib
cd mjpg-streamer-code-182/mjpg-streamer
cp *.so /home/fengjunhui/source/rootfs/mjpg-streamer -arf
cp mjpg-stream /home/fengjunhui/source/rootfs/mjpg-streamer -arf

```

将源码目录中的 start.sh 和目录 www 拷贝到~/source/rootfs/mjpg-streamer/目录下，然后就可以测试啦

```

sudo cp start.sh www/ ~/source/rootfs/mjpg-streamer/ -arf

```

7、获得帮助

获得帮助的方法在 start.sh 里有说明，以下举一个例子，运行./mjpg_streamer -help 它会打印出以下帮助，可根据需要设置参数

```

The following parameters can be passed to this plugin:
[-d | -device ].....: video device to open (your camera)
[-r | -resolution ]...: the resolution of the video device,
can be one of the following strings
QSIF QCIF CGA QVGA CIF VGA
SVGA XGA SXGA
or a custom value like the following
example: 640x480
[-f | -fps ].....: frames per second
[-y | -yuv ].....: enable YUYV format and disable MJPEG mode
[-q | -quality ].....: JPEG compression quality in percent
(activates YUYV format, disables MJPEG)
[-m | -minimum_size ].: drop frames smaller then this limit, useful
if the webcam produces small-sized garbage frames
may happen under low light conditions
[-n | -no_dynctrl ]...: do not initialize dynctrls of Linux-UVC driver

```

```
[-1 | -led ].....: switch the LED "on", "off", let it "blink" or  
leave  
it up to the driver using the value "auto"
```