

视频监控的实现

-- mjpg-streamer 的实现

作者：深圳百问网科技有限公司 万勇 www.100ask.net, 100ask.taobao.com

1. 环境

硬件：电脑、JZ2440 开发板、UVC 摄像头(推荐使用) 系统：Ubuntu9.10

// 即光盘提供的那个虚拟机系统

Uboot：u-boot-2012.04.01

// 即毕业班移植新 uboot 那个

Kernel：linux-3.4.2

// 即毕业班移植新内核那个文件系统：

fs_mini_mdev_new.tar.bz2

基本概念：

- ① UVC 指 USB Video Class，UVC 摄像头的简单判断标准就是接到 Windows 电脑上后，不用安装驱动程序就可以使用
- ② USB 摄像头输出的数据有多种格式，比如原始数据 RGB 或 YUV 格式，压缩的 MJPEG 格式。使用网络传输视频时，应该传输压缩格式的，否则传输的数据量会非常大。
S3C2440 的主频只有 400M，如果使用的摄像头不支持 MJPEG 输出，就需要使用软件(mjpg-streamer)来压缩，这极大的耗费 CPU 资源，导致远程视频不流畅

摄像头选购指南：

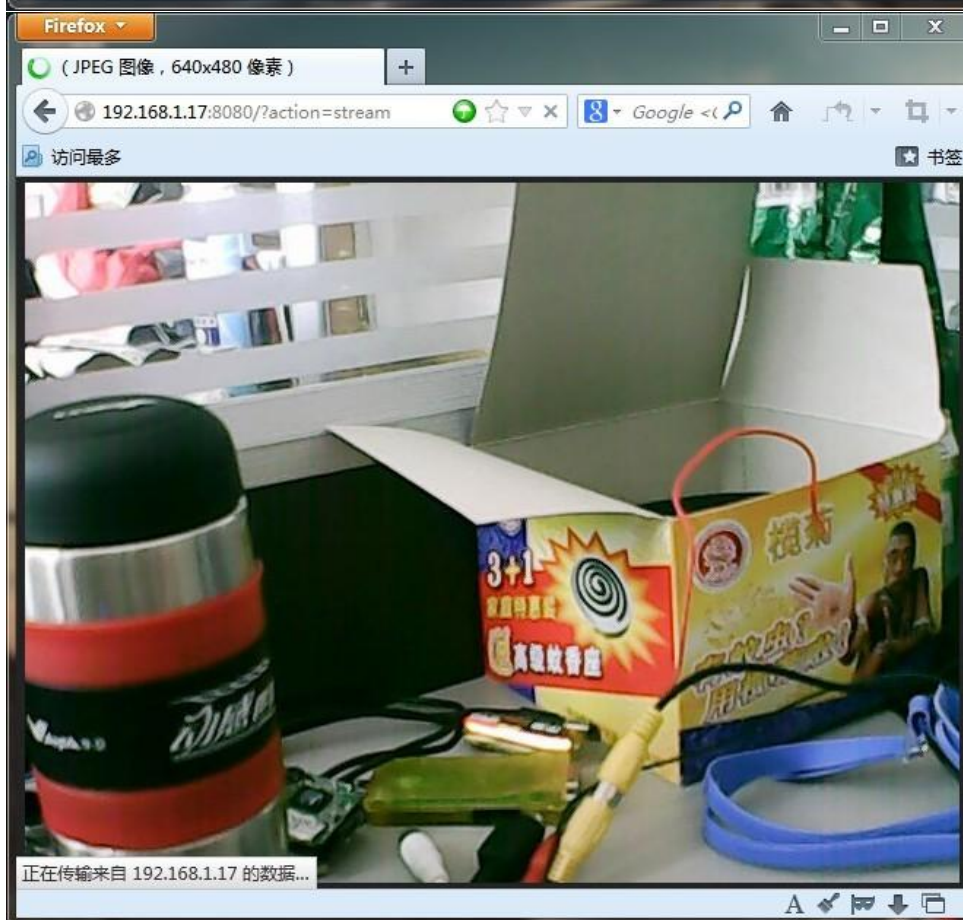
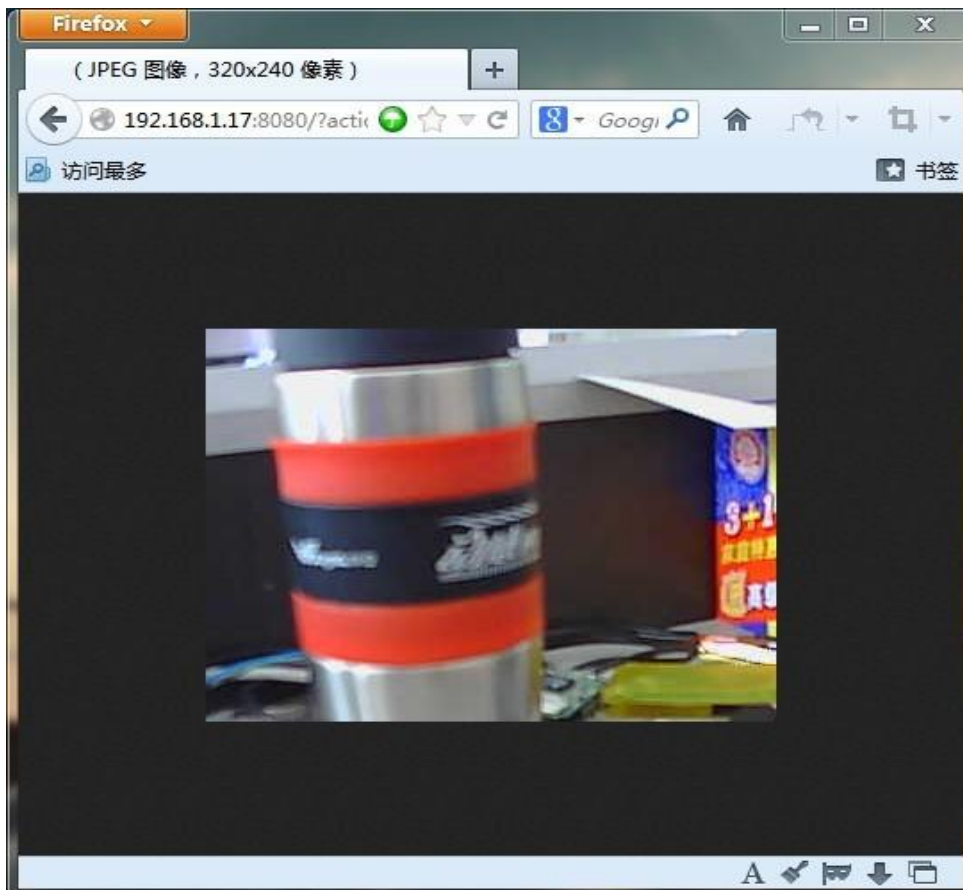
- ① 如果只是基于原理性的学习，不考虑性能，可以购买一般的 UVC 摄像头，价格在 20~50 元之间，它输出原始数据，不支持输出 MJPEG 数据。
- ② 如果能让远程视频很流畅，就需要选购支持 MJPEG 的摄像头，这些摄像头既可以输出 MJPEG 数据，也可以输出原始数据。我测试过“蓝色妖姬 S3”，它的价格在 70~80 之间。市面上还有很多支持 MJPEG 的摄像头，价格基本上百。可以参考这个贴子：<http://www.right.com.cn/forum/thread-74690-1-1.html>
- ③ 我们也将制作一个摄像头模块，它的硬件框架大概是这样：

CMOS 摄像头==>视频控制器芯片==>USB 口 在第 2 个项目里，这个模块可以接到开发板的 USB 口去作为 USB 摄像头；也可以把上面的 CMOS 摄像头取下来，接到开发板上的 CMOS 插槽去。它支持 JZ2440、TQ2440、MINI2440。

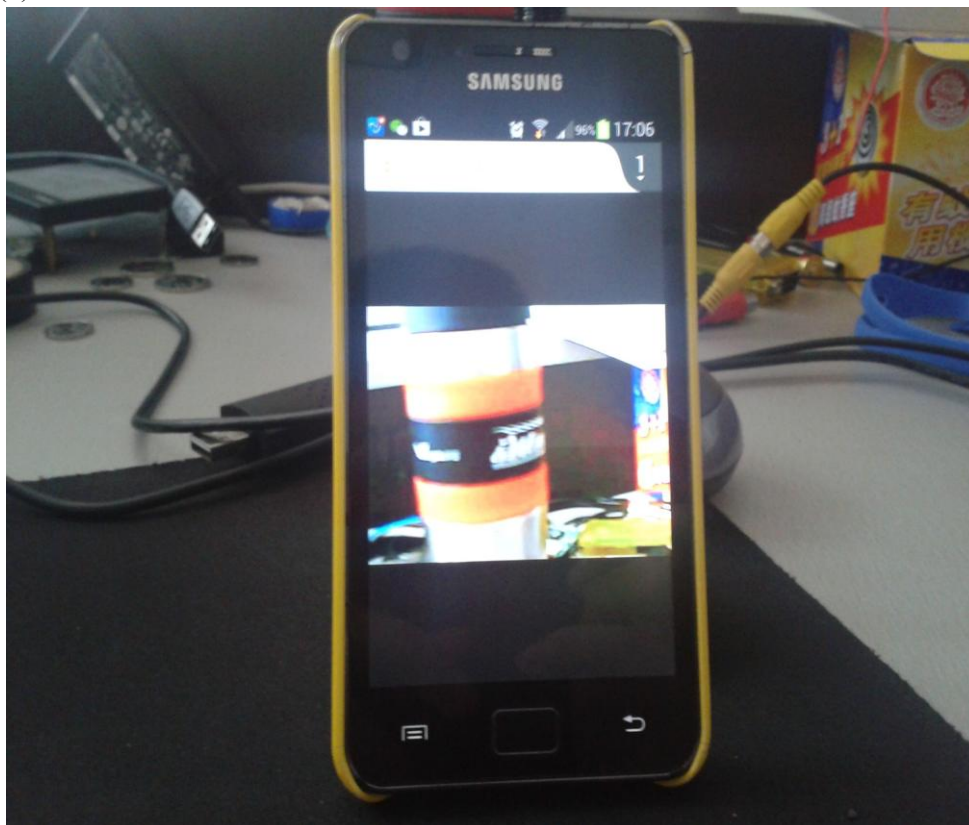
此模块正在设计中，面市时可以从 100ask.taobao.com 买到。

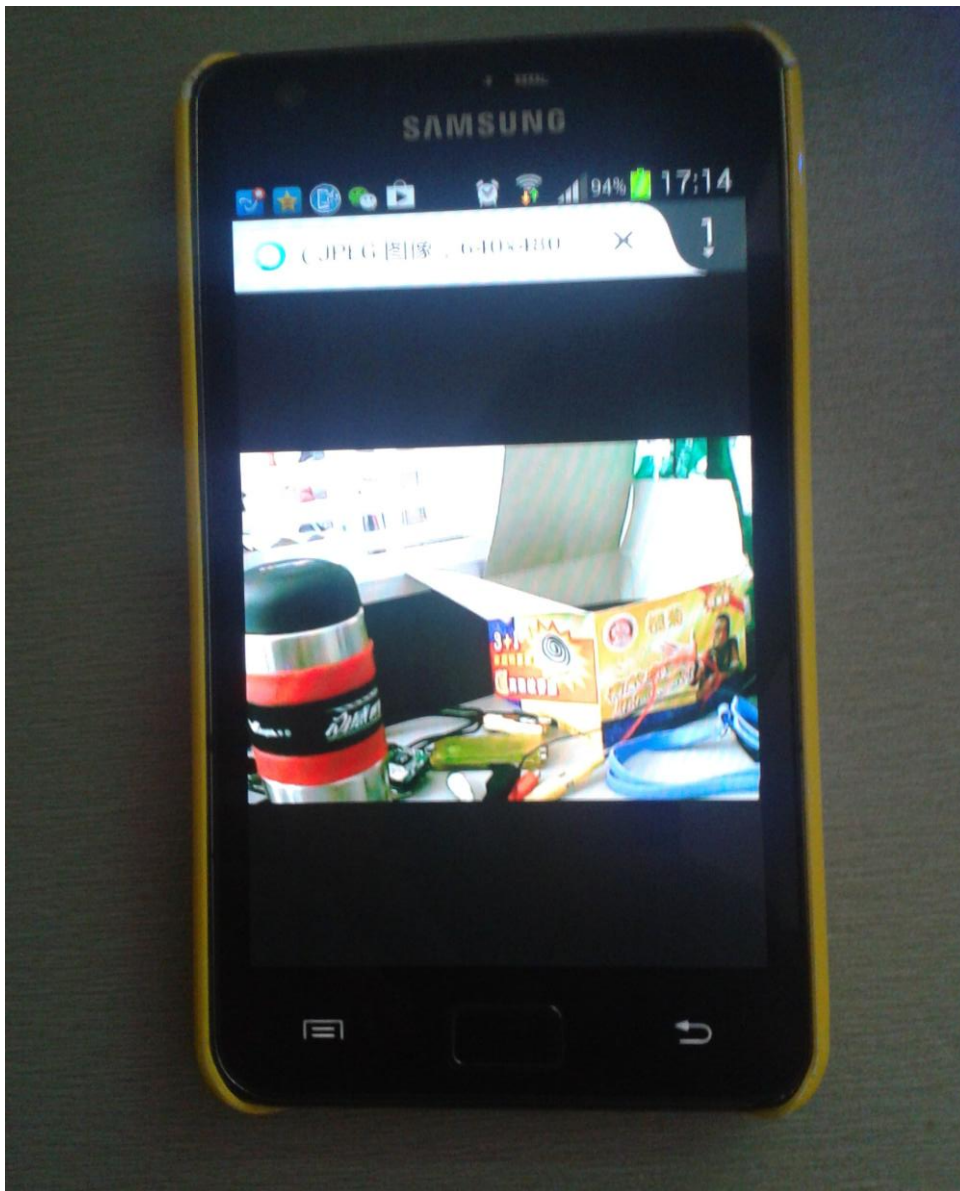
2. 效果图

(1). 电脑上的效果图：



(2). 手机上的效果图:





3. 移植步骤

搭建环境:

(1) 配置编译内核

```
tar cd patch -p1 < ../linux-
```

```
3.4.2_100ask.patch cp
```

```
config_ok ./config make menuconfig
```

// 配置内核，使其支持 UVC 摄像头，如下

-> Device Drivers

<*> Multimedia support --->

<*> Video For Linux

[*] Video capture adapters (NEW) --->

[*] V4L USB devices (NEW) --->

<*> USB Video Class (UVC)

// 配置内核，使其支持 USB 总线驱动，如下

Device Drivers --->

[*] USB support --->

{*} Support for Host-side USB

[*] USB device filesystem (DEPRECATED)

[*] USB device class-devices (DEPRECATED)

<*> OHCI HCD support

<*> USB Mass Storage support

[*] HID Devices --->

{*} Generic HID support

[*] /dev/hidraw raw HID device support

SCSI device support --->

<*> SCSI device support

[*] legacy /proc/scsi/ support

<*> SCSI disk support

<*> SCSI tape support

make uImage

(2) 网络挂载文件系统

```
set ipaddr 10.10.10.122 && set serverip 10.10.10.200 && tftp 0x30000000 uImage set bootargs
root=/dev/nfs nfsroot=192.168.1.105:/work/nfs_root/first_fs ip=192.168.1.17 console=
ttySAC0,115200 bootm 0x30000000
```

注意：该步骤需要网卡的支持，mjpg-streamer 的测试也需要网卡的支持，因此你的开发板必须要有网卡驱动。怎么移植网卡，请看韦东山二期视频。

移植 mjpg-streamer：分析过 mjpg-streamer 的代码的人一定会发现，mjpg-streamer 是会依赖于 libjpeg！因此我们需要先移植 libjpeg。

(1) libjpeg 的移植(韦东山第三期视频的第一个项目中已经讲过)：

```
tar xzvf libjpeg-turbo-1.2.1.tar.gz
```

```
cd libjpeg-turbo-1.2.1 mkdir tmp
```

```
./configure --prefix=/work/cross_mjpeg/libjpeg-turbo-1.2.1/tmp/ --host=arm-linux make
```

```
make install
```

```
cp libjpeg-turbo-1.2.1/tmp/lib/*so* /work/nfs_root/fs_mini_mdev_new/lib/ -d
```

// 将编译出来的结果
// 拷贝到文件系统中

(2) mjpg-streamer 的移植：

```
tar xvf mjpg-streamer-r63.tar.gz cd
```

```
mjpg-streamer-r63
```


修改所有的 Makefile

--(1) 将 CC=gcc 修改为 CC=arm-linux-gcc

--(2) 修改 plugins/input_uvc/Makfile a. 将

CFLAGS += -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC 改为

CFLAGS += -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -I /work/cross_mjpeg

/libjpeg-turbo-1.2.1/tmp/include 注意:

-I /work/cross_mjpeg/libjpeg-turbo-1.2.1/tmp/include // 是编译 libjpeg 生成的文件

b. 将 input_uvc.so: \$(OTHER_HEADERS) input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo

\$(CC) \$(CFLAGS) -ljpeg -o

\$@ input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo

改为

input_uvc.so: \$(OTHER_HEADERS) input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo

\$(CC) \$(CFLAGS) -ljpeg -L

/work/cross_mjpeg/libjpeg-turbo-1.2.1/tmp/lib -o

\$@ input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo

make mkdir mjpeg cp

mjpg_streamer mjpeg cp

*.so mjpeg

cp -rf mjpeg/ /work/nfs_root/first_fs // 将编译出的结果拷贝到根文件系统中

测试(都在开发板上运行):

1). 接上 UVC 摄像头到开发板;

2). 查看设备节点

ls /dev/vid*

便可以看到我们摄像头对应的设备节点 video#

3). 运行 mjpg_streamer

cd mjpeg/

./mjpg_streamer -i "/input_uvc.so -y" -o "/output_http.so -w ./www"

./mjpg_streamer -i "/input_uvc.so" -o "/output_http.so -w ./www"

注意:

mjpg_streamer 的使用可以通过加入 -h 选项来查看帮助说明!!

4). 电脑或手机接上 wifi 信号或者通过网线连接, 然后在火狐浏览器或者 google 浏览器上输入如下

URL: <http://192.168.1.17:8080/?action=stream>

注意:

IP 地址是你开发板的 IP, 请根据自己的情况修改!!

