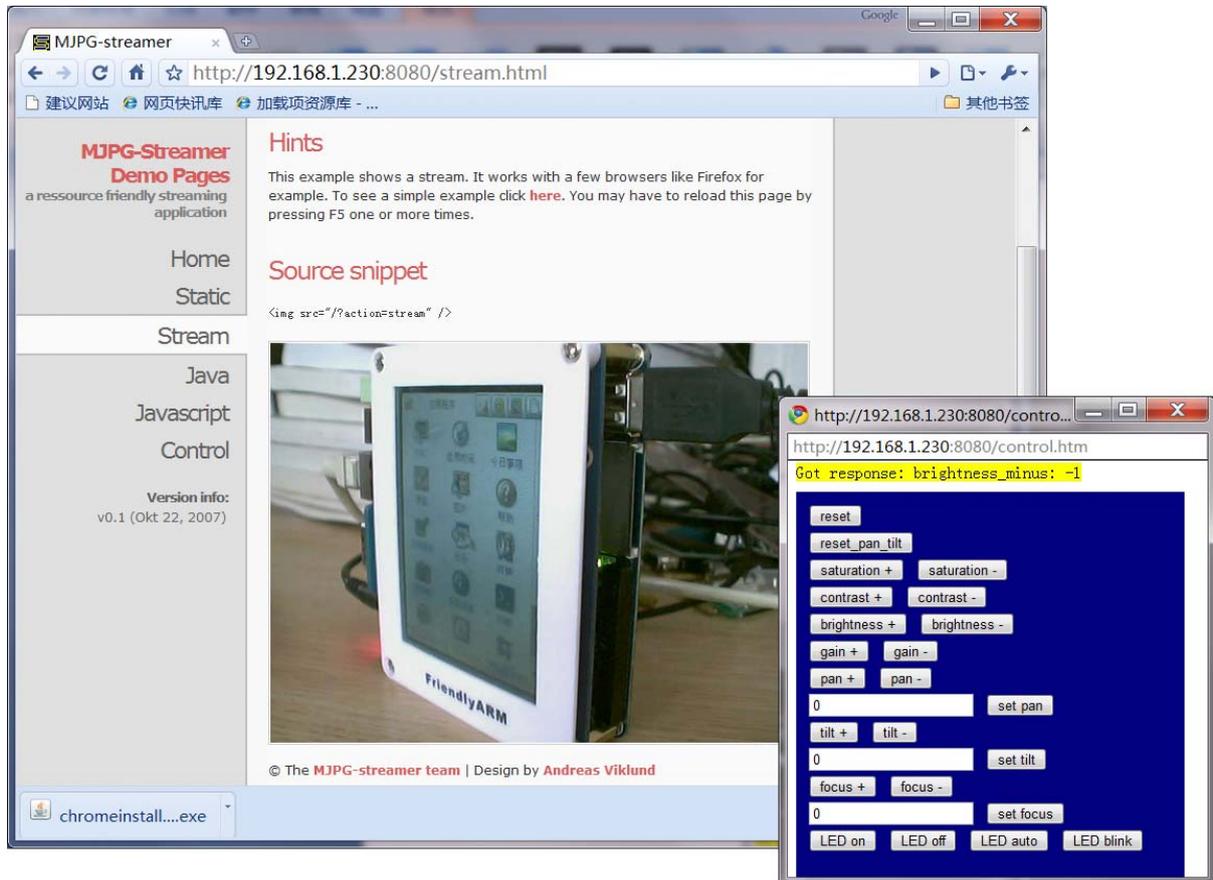


基于 Mini2440 通过网络浏览和控制 CMOS 或 USB 摄像头

本文档版本日期：2010-4-5



说明：本文档关键内容主要来自于网络，敬请自由传播浏览，友善之臂保留本文档的解释权及修改权。

更新说明

2010-4-5: 增加了某些 USB 摄像头不被支持的原因解释

2010-4-1: 更改了目录标题序号，以和实际目录对应。

2010-3-27: 本文档创建



目录

1 简介	2
2 快速上手体验	2
2.1. 环境平台	2
2.2 在 mini2440 上安装 mjpeg-stream.....	3
2.3 通过网页浏览 CMOS 摄像头	3
2.4 通过网页浏览和控制 USB 摄像头.....	5
3 关于下载和编译源代码	9

1 简介

虽然我们已经提供了基于 Qtopia 的摄像头预览和拍照实用测试程序(不开源),但依然有很多网友和用户希望能够通过网络控制和浏览摄像头。互联网的力量十分伟大,现在已经有国外的爱好者在 mini2440 上把这个想法付诸实践并实现了,并为此创建了一个 google code 项目,网址是: <http://code.google.com/p/mjpg-streamer-mini2440/>,我们经过简单测试使用,觉得这个东西十分简单易用,特此介绍给大家。

Mjpeg-stream 是一个开源软件,用于把 mjpeg 格式的文件变成流媒体输出,这里作者已经把它移植到 mini2440,并做好了相应的启动运行命令,因此你只需下载下来使用它就行了。

2 快速上手体验

2.1. 环境平台

目标板: mini2440/micro2440

开发板上运行的系统: 内核 Linux-2.6.32.2, 默认文件系统 root_qtopia(不分 nand flash 容量类型)

主机: Windows7 + Chrome 浏览器(可能需要安装 Java 组件)

说明: 默认的 Linux-2.6.32.2 内核已经支持 CMOS 摄像头, 及多种型号的 USB 摄像头, 因此无需重新配置编译。



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

2.2 在 mini2440 上安装 mjpeg-stream

Step1: 获取要安装的文件 mjpeg-stream

文件名: mjpg-streamer-mini2440-bin-r6.tar.gz

获取途径: (1)使用附件中的可执行文件(2)自行下载

下载地址:

<http://mjpg-streamer-mini2440.googlecode.com/files/mjpg-streamer-mini2440-bin-r6.tar.gz>

Step2: 在 mini2440 上安装 mjpeg-stream

使用优盘或者 ftp 等工具把 mjpeg-stream 复制到开发板, 在此假定复制到了/home/plg 目录
在控制台终端执行如下命令:

```
#mkdir /mjpg-streamer ;创建一个独立目录用于测试
```

```
#cd /mjpg-streamer ;进入该测试目录
```

```
#tar xvzf mjpg-streamer-mini2440-bin-r6.tar.gz ;解压安装
```

至此, mjpeg-stream 就已经在开发板上安装好了。

2.3 通过网页浏览 CMOS 摄像头

Step1: 连接好硬件

把 CMOS 摄像头插装到开发板上, 可以先使用开发板自带的“CMOS 摄像头”程序测试该摄像头可用, 然后重启开发板。

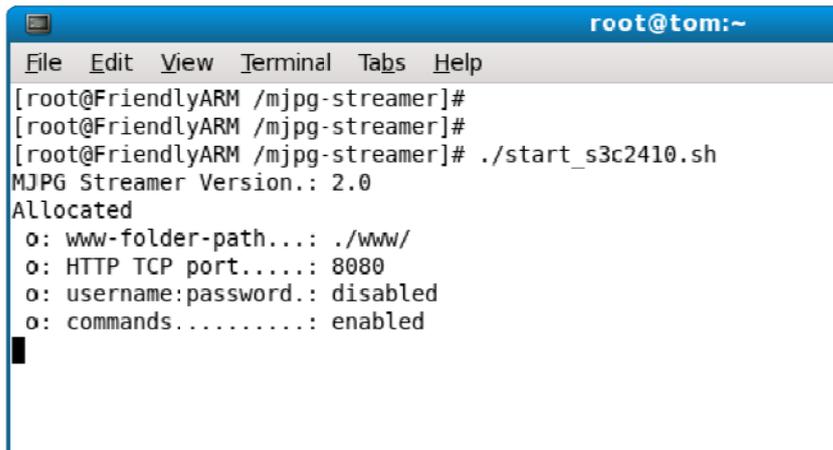
Step2: 启动 mjpeg-stream

在命令行终端执行以下命令:

```
#cd /mjpg-streamer ;进入测试目录
```

```
#./start_s3c2410.sh ;执行测试程序
```

此时可以在终端看到如图输出信息, 这说明已经成功启动 mjpeg-stream 程序

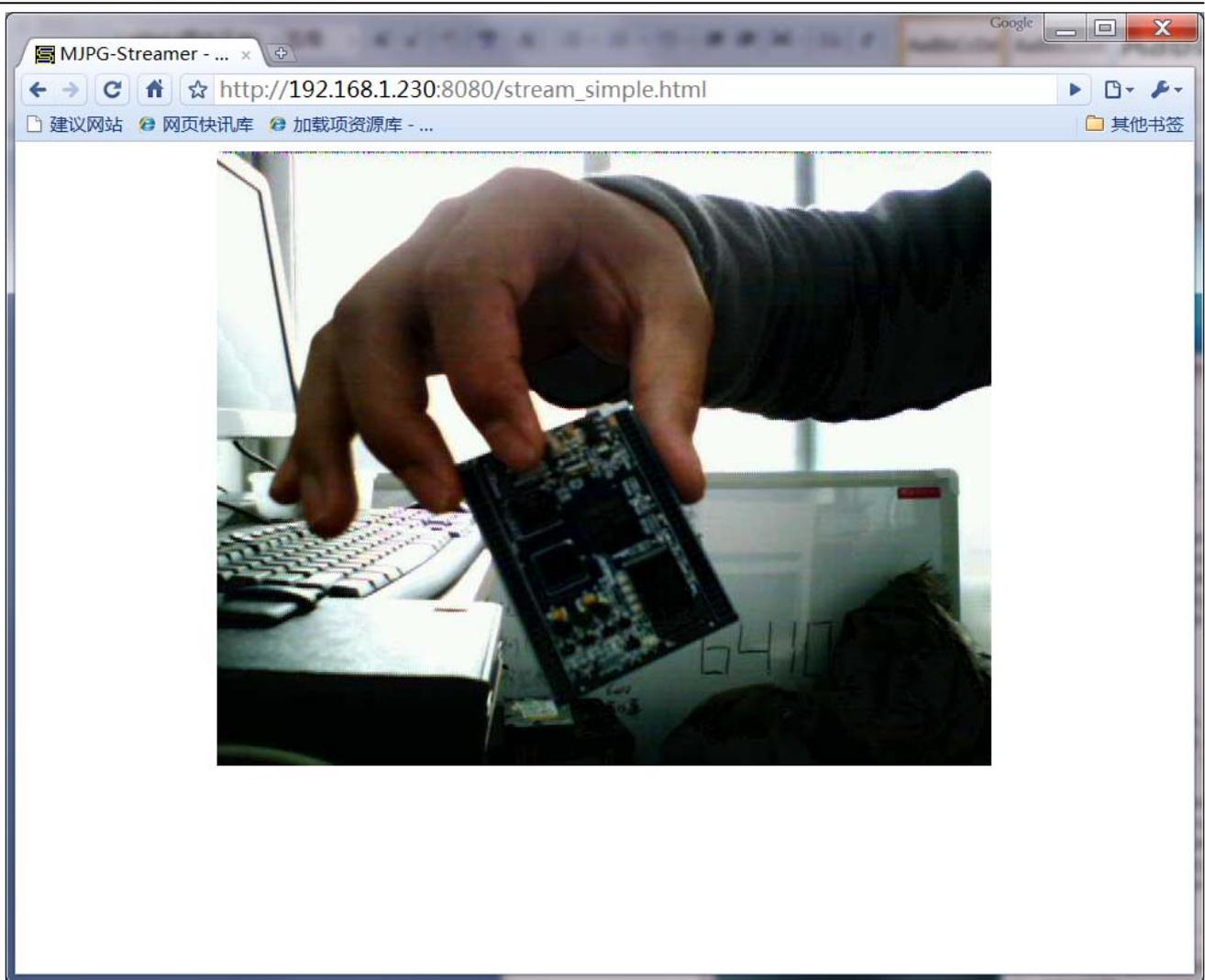


```
root@tom:~  
File Edit View Terminal Tabs Help  
[root@FriendlyARM /mjpg-streamer]#  
[root@FriendlyARM /mjpg-streamer]#  
[root@FriendlyARM /mjpg-streamer]# ./start_s3c2410.sh  
MJPG Streamer Version.: 2.0  
Allocated  
o: www-folder-path....: ./www/  
o: HTTP TCP port.....: 8080  
o: username:password.: disabled  
o: commands.....: enabled
```

Step3: 通过浏览器动态预览 CMOS 摄像头

在浏览器中输入：http://192.168.1.230:8080/stream_simple.html 即可看到摄像头的动态画面了，如图。

说明：开发板默认的 IP 地址是 192.168.1.230 ，你可以根据实际的网络参数输入相应的 ip 地址。



2.4 通过网页浏览和控制 USB 摄像头

Step1: 连接好硬件

把 USB 摄像头插装到开发板上，可以先使用开发板自带的“USB 摄像头”程序测试该摄像头可用，然后重启开发板。

Step2: 启动 mjpeg-stream

在命令行终端执行以下命令：

```
#cd /mjpeg-streamer ;进入测试目录
```

```
#!/start_uvc.sh ;执行测试程序
```

此时可以在终端看到如图输出信息，这说明已经成功启动 mjpeg-stream 程序

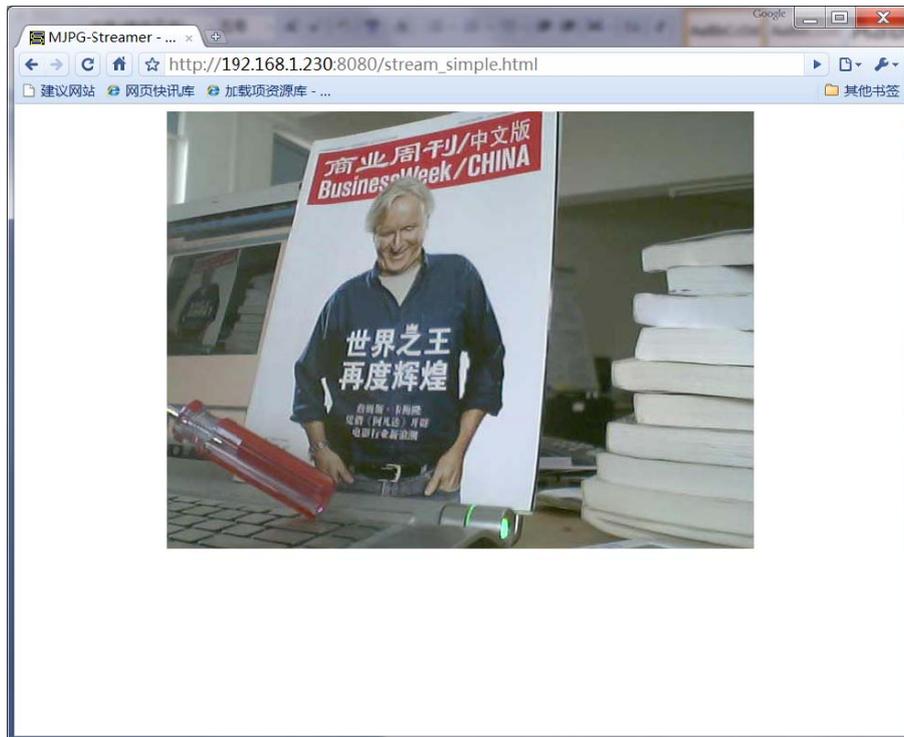
```
force cancelation of threads and cleanup rresources
o: cleaning up rresources allocated by server thread #00
[root@FriendlyARM /mjpg-streamer]# ./start_s3c2410.sh done

[root@FriendlyARM /mjpg-streamer]# ./start_uvc.sh
MJPEG Streamer Version.: 2.0
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 640 x 480
i: Frames Per Second.: 5
i: Format.....: MJPEG
o: www-folder-path...: ./www/
o: HTTP TCP port.....: 8080
o: username:password.: disabled
o: commands.....: enabled
```

Step3: 通过浏览器动态预览 USB 摄像头

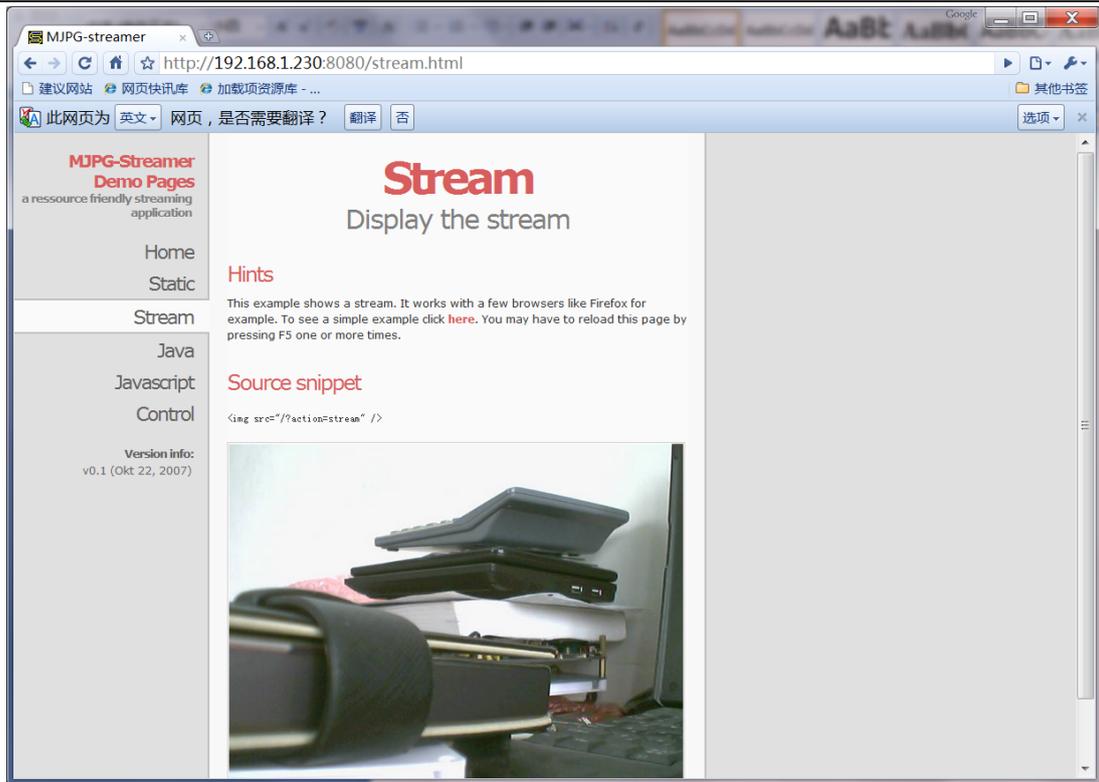
在浏览器中输入：http://192.168.1.230:8080/stream_simple.html 即可看到摄像头的动态画面了，如图。

说明：开发板默认的 IP 地址是 192.168.1.230，你可以根据实际的网络参数输入相应的 ip 地址。因为大部分 USB 摄像头的分辨率要高一些，因此呈现的图像也会清晰很多。



Step4: 通过浏览器控制 USB 摄像头

在浏览器中输入：<http://192.168.1.230:8080/stream.html> 出现如下界面



点页面左侧的“Control”会出来一个控制页面窗口，如图



通过页面上的提示，可以控制 USB 摄像头的对比度，亮度等参数，用户可以自行测试一下。

2.5 为什么我的 USB 摄像头没有显示

如果你使用的 USB 摄像头在使用时出现了如下信息：

```
[root@FriendlyARM /mjpg-streamer]# ./start_uvc_yuv.sh
MJPEG Streamer Version.: 2.0
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 640 x 480
i: Frames Per Second.: 5
i: Format.....: YUV
i: JPEG Quality.....: 80
Pixel format is unavailable, using JPEG
should never arrive exit fatal !!
i: init_VideoIn failed
```

这说明，mjpeg 软件并不支持你所使用的 USB 摄像头类型。解决方法有两种：

- (1) 你可以更改其他型号的 USB 摄像头试试
- (2) 自行更改 mjpeg 源代码以支持你所使用的 USB 摄像头驱动程序接口

解释：虽然 Linux-2.6.32 等高版本的内核已经支持“万能”USB 摄像头驱动，但每个驱动向上层提供的图像解码接口是不同的，而 mjpeg 软件仅能识别一部分型号的 USB 摄像头，而我们提供的 Qtopia 版本的 USB 摄像头动态预览程序，则根据用户的不断反馈增加了更多的 USB 摄像头向上层提供的图像解码接口支持，这也就是我们为什么会在该程序中添加了这样一个提示：



目的就是让用户把所使用的 USB 摄像头样品寄给我们，以便在我们的程序中增加该型号 USB 摄像头驱动中的图像解码支持。对于 mjpeg 软件，我们则不提供这样的升级改进，因为 mjpeg 软件并不是我们开发和维护的。

3 关于下载和编译源代码

在 <http://code.google.com/p/mjpg-streamer-mini2440/source/checkout> 可以下载到以上程序的源代码，附件是我已经下载好的源代码压缩文件。

文件名：**mjpg-streamer-mini2440-read-only.tgz**

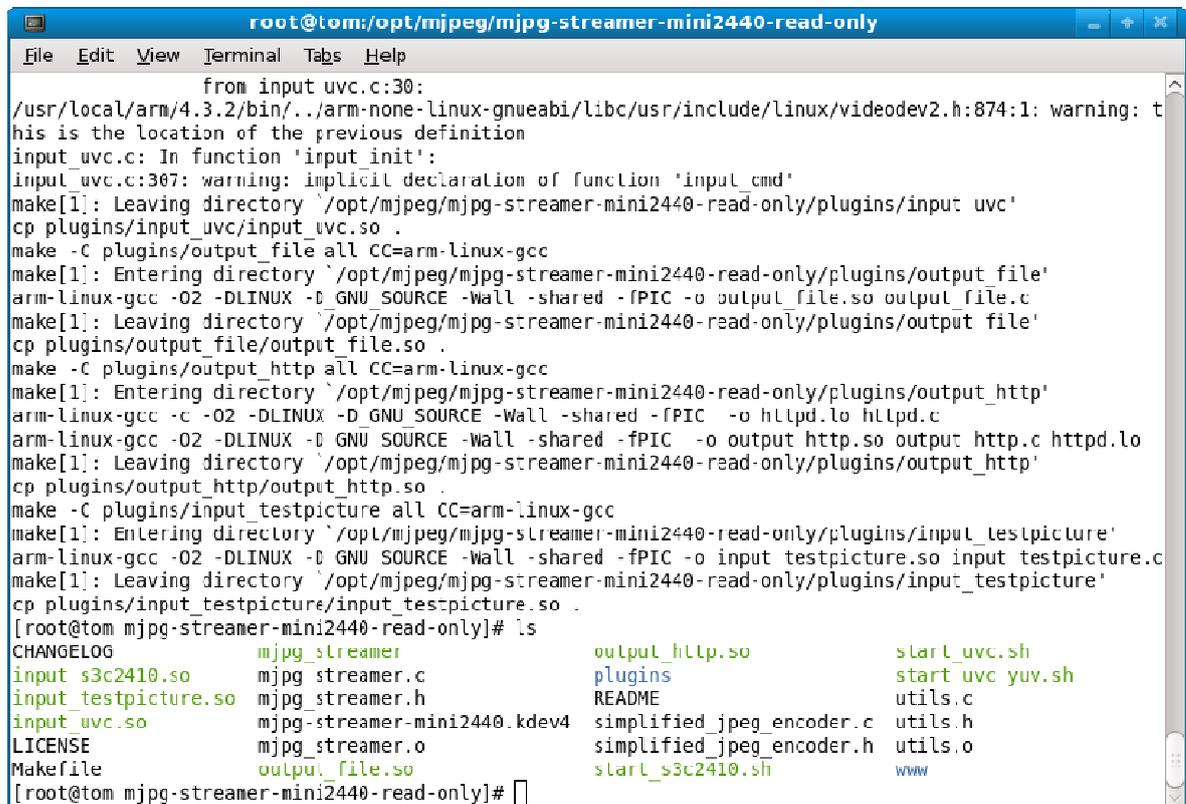
因为作者已经做好了所有移植等工作，编译该源代码十分简单。把它解压到任意目录，如/opt 目录：

```
#cd /opt ;进入 opt 目录
```

```
#tar xvzf mjpg-streamer-mini2440-read-only.tgz ;解压
```

```
#make CC=arm-linux-gcc ;编译
```

最后会生成 mjpeg-stream 可执行文件，如图：



```
root@tom:/opt/mjpeg/mjpg-streamer-mini2440-read-only
File Edit View Terminal Tabs Help
      from input_uvc.c:30:
/usr/local/arm/4.3.2/bin/../arm-none-linux-gnueabi/libc/usr/include/linux/videodev2.h:874:1: warning: this is the location of the previous definition
input_uvc.c: In function 'input_init':
input_uvc.c:367: warning: implicit declaration of function 'input_cmd'
make[1]: Leaving directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/input_uvc'
cp plugins/input_uvc/input_uvc.so .
make -C plugins/output_file all CC=arm-linux-gcc
make[1]: Entering directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/output_file'
arm-linux-gcc -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o output_file.so output_file.c
make[1]: Leaving directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/output_file'
cp plugins/output_file/output_file.so .
make -C plugins/output_http all CC=arm-linux-gcc
make[1]: Entering directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/output_http'
arm-linux-gcc -c -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o httpd.lo httpd.c
arm-linux-gcc -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o output_http.so output_http.c httpd.lo
make[1]: Leaving directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/output_http'
cp plugins/output_http/output_http.so .
make -C plugins/input_testpicture all CC=arm-linux-gcc
make[1]: Entering directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/input_testpicture'
arm-linux-gcc -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o input_testpicture.so input_testpicture.c
make[1]: Leaving directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/input_testpicture'
cp plugins/input_testpicture/input_testpicture.so .
[root@tom mjpg-streamer-mini2440-read-only]# ls
CHANGELOG      mjpg_streamer      output_http.so      start_uvc.sh
input_s3c2410.so  mjpg_streamer.c    plugins              start_uvc_yuv.sh
input_testpicture.so  mjpg_streamer.h  README              utils.c
input_uvc.so      mjpg-streamer-mini2440.kdev4  simplified_jpeg_encoder.c  utils.h
LICENSE          mjpg_streamer.o   simplified_jpeg_encoder.h  utils.o
Makefile         output_file.so    start_s3c2410.sh     www
```

注意：请确认你使用的编译器是友善之臂提供的 arm-linux-gcc 4.3.2，运行 arm-linux-gcc -v 命令可以检查交叉编译器的版本，如图：



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

```
root@tom:/opt/mjpeg/mjpg-streamer-mini2440-read-only
File Edit View Terminal Tabs Help
make -C plugins/input_testpicture all CC=arm-linux-gcc
make[1]: Entering directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/input_testpicture'
arm-linux-gcc -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o input_testpicture.so input_testpicture.c
make[1]: Leaving directory `/opt/mjpeg/mjpg-streamer-mini2440-read-only/plugins/input_testpicture'
cp plugins/input_testpicture/input_testpicture.so .
[root@tom mjpg-streamer-mini2440-read-only]# ls
CHANGELOG          mjpg_streamer          output_http.so        start_uvc.sh
input_s3c2410.so   mjpg_streamer.c        plugins               start_uvc_yuv.sh
input_testpicture.so mjpg_streamer.h        README                utils.c
input_uvc.so       mjpg-streamer-mini2440.kdev4 simplified_jpeg_encoder.c  utils.h
LICENSE           mjpg_streamer.o        simplified_jpeg_encoder.h  utils.o
Makefile          output_file.so         start_s3c2410.sh      www
[root@tom mjpg-streamer-mini2440-read-only]# arm-linux-gcc
arm-linux-gcc      arm-linux-gcc-4.3.2
[root@tom mjpg-streamer-mini2440-read-only]# arm-linux-gcc -v
Using built-in specs.
Target: arm-none-linux-gnueabi
Configured with: /scratch/julian/lite-respin/linux/src/gcc-4.3/configure --build=i686-pc-linux-gnu --host=i686-pc-linux-gnu --target=arm-none-linux-gnueabi --enable-threads --disable-libmudflap --disable-libssp --disable-libstdcxx-pch --with-gnu-as --with-gnu-ld --enable-languages=c,c++ --enable-shared --enable-symvers=gnu --enable-__cxa_atexit --with-pkgversion='Sourcery G++ Lite 2008q3-72' --with-bugurl=http://support.codesourcery.com/GNUToolchain/ --disable-nls --prefix=/opt/codesourcery --with-sysroot=/opt/codesourcery/arm-none-linux-gnueabi/libc --with-build-sysroot=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/libc --with-gmp=/scratch/julian/lite-respin/linux/obj/host-libs-2008q3-72-arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --with-mpfr=/scratch/julian/lite-respin/linux/obj/host-libs-2008q3-72-arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --disable-libgomp --enable-poison-system-directories --with-build-time-tools=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/bin --with-build-time-tools=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/bin
Thread model: posix
gcc version 4.3.2 (Sourcery G++ Lite 2008q3-72)
[root@tom mjpg-streamer-mini2440-read-only]#
```