



基于 TQ2440 开发板的 Linux- 2.6.30.4 的网络摄像机的实现

广州天嵌计算机科技有限公司荣誉出品

首发网站: www.embedsky.net



版权声明

本手册版权归属广州天嵌计算机科技有限公司（以下简称“天嵌科技”）所有，并保留一切权力。非经天嵌科技同意（书面形式），任何单位及个人不得擅自摘录本手册部分或全部内容，违者（我们）公司将追究其法律责任。

EmbedSky

天嵌科技



目录

版权声明.....	1
目录.....	2
Step 1: Linux-2.6.30.4 内核配置.....	3
Step 2: 准备工作.....	4
Step 3: 移植 SDL 库.....	4
Step 4: 编译 mjpg-streamer.....	5
Step 5: 编译 uvc-streamer.....	6
Step 6: 测试网络摄像机.....	6
使用 mjpg-streamer 测试 USB 摄像头:	6
使用 mjpg-streamer 测试 CMOS 摄像头:	8
使用 uvc-stream 测试 USB 摄像头:	10

Embedsky

天嵌科技



Step 1: Linux-2.6.30.4 内核配置

在 Linux-2.6.30.4 的内核中，我们没有从外部移植 USB 摄像头的驱动，而直接采用了内核自带的 USB 摄像头驱动，那么我们以前针对 USB 摄像头做的网络摄像头控制程序就不能使用了（原来的 servfox 代码需要修改），下面我们使用新的方法实现网络摄像机功能。

注意：在我们提供的默认配置单中只添加了对 ZC301 系列摄像头的配置，如果您要添加对别的摄像头的支持，只需要在如下的配置单中选择上对应的配置选项即可。

配置单如下：（红色部分所示）

```
Device Drivers --->
Multimedia devices --->
*** Multimedia core support ***
  <*> Video For Linux
    [*] Enable Video For Linux API 1 (DEPRECATED)
    [*] Video capture adapters --->
      --- Video capture adapters
        <*> OV9650 Drivers for EmbedSky SKY2440/TQ2440 Board
          [*] V4L USB devices --->
            --- V4L USB devices
              <*> USB Video Class (UVC)
                [*] UVC input events device support
              <*> GSPCA based webcams --->
                --- GSPCA based webcams
                  <> ALi USB m5602 Camera Driver
                  <> STV06XX USB Camera Driver
                  <> Conexant Camera Driver
                  <> Etoms USB Camera Driver
                  <> Fujifilm FinePix USB V4L2 driver
                  <> Mars USB Camera Driver
                  <> Mars-Semi MR97310A USB Camera Driver
                  <> OV519 USB Camera Driver
                  <> OV534 USB Camera Driver
                  <> Pixart PAC207 USB Camera Driver
                  <> Pixart PAC7311 USB Camera Driver
                  <> SONIX Bayer USB Camera Driver
                  <> SONIX JPEG USB Camera Driver
                  <> SPCA500 USB Camera Driver
                  <> SPCA501 USB Camera Driver
                  <> SPCA505 USB Camera Driver
                  <> SPCA506 USB Camera Driver
                  <> SPCA508 USB Camera Driver
                  <> SPCA561 USB Camera Driver
```



<> SQ Technologies SQ905 based USB Camera Driver
<> SQ Technologies SQ905C based USB Camera Driver
<> Syntek DV4000 (STK014) USB Camera Driver
<> SUNPLUS USB Camera Driver
<> T613 (JPEG Compliance) USB Camera Driver
<> TV8532 USB Camera Driver
<> VC032X USB Camera Driver
<*> ZC3XX USB Camera Driver

<*> USB ZC0301[P] Image Processor and Control Chip support

Step 2: 准备工作

我们使用 4.3.3 的交叉编译器，我们需要移植 SDL 库（<http://soft.embedsky.net/viewthread.php?tid=630&extra=page%3D1> 提供的下载链接中 EABI_4.3.3_EmbedSky_20090812.tar.bz2 这个包已经添加了 SDL 库），还需要 mjpg_streamer 源码。

首先我们需要下载一个名为：mjpg_streamer 的源码包，下载地址：<http://mjpg-streamer.svn.sourceforge.net/viewvc/mjpg-streamer.tar.gz?view=tar>（在光盘的“Linux 资源\Linux 源码包\2.6.30.4”目录下包含了该源码），然后我们将其复制到 PC 的 Linux 系统中，然后解压源码，使用命令：“#tar xvfz mjpg-streamer.tar.gz -C /opt/EmbedSky/apps/”，然后进到源码中“#cd /opt/EmbedSky/apps/mjpg-streamer”。

然后把源码中的所有的“Makefile”文件的 CC=gcc 改为 CC=arm-linux-gcc，然后就可以编译了。

也可以直接从我们网站下载已经修改过后的 mjpg-streamer 源码，下载地址：http://soft.embedsky.net/files/linux_src/mjpg-streamer.tar.bz2，解压命令用：“#tar xvfj mjpg-streamer.tar.bz2 -C /”，解压后的源码在“/opt/EmbedSky/apps/WebCam/mjpg-streamer/”目录下。

说明：推荐使用我们修改后的源码，因为第一个链接可能出现版本更新导致编译不过去的情况，并且下面的介绍完全依赖我们修改后的源码进行讲解。

在“/opt/EmbedSky/apps/WebCam/mjpg-streamer/”源码下面有 3 个目录：mjpeg-client、mjpg-streamer 和 uvc-streamer。

- mjpeg-client 目录下的 bin/win32_upx/目录下存在一个名为：viewer.exe 的 Windows 平台的应用程序，我们将 win32_upx 目录复制到 Windows 平台，即可执行 viewer.exe 程序，该程序是作为 PC 端通过网络查看摄像头采集图像的播放器；
- mjpg-streamer 目录下提供了 mjpg-streamer 的可执行程序 and 各个输入输出设备组件；
- uvc-streamer 目录下提供 uvc-streamer 的可执行程序。

Step 3: 移植 SDL 库

我们从这里下载：<http://www.libsdl.org/release/SDL-1.2.13.tar.gz> 的源码包，然后将其复制到 PC 的 Linux 中，然后解压源码：“#tar xvfz SDL-1.2.13.tar.gz -C /opt/EmbedSky/apps/”，解压后的源码在“/opt/EmbedSky/apps/SDL-1.2.13/”目录下。

在刚刚解压的源码下新建一个“#mkdir __install”的目录用于存放好的库和头文件；

使用如下命令进行 SDL 的配置：“#./configure --prefix=/opt/EmbedSky/apps/SDL-1.2.13/__install --



```
disable-video-nanox --enable-video-qtopia --disable-video-photon --disable-video-direct --disable-video-ggi --  
disable-video-svga --disable-video-aalib --disable-video-dummy --disable-video-dga --disable-arts --disable-esd --  
disable-alsa --disable-video-x11 --disable-nasm --disable-debug --disable-joystick-amigaos --disable-joystick-beos  
--disable-joystick-bsd --disable-joystick-darwin --disable-joystick-dc --enable-joystick-linux --disable-joystick-  
macos --disable-joystick-mint --disable-joystick-win32 --disable-joystick --host=arm-linux --build=i386”
```

注意：前面那个配置参数是首先使用命令“`#!/configure --help`”查看配置选项后，经过试验和经验的判断得到的。

使用命令“`#make;make install`”编译并安装 SDL 库；

最后进到“`install`”目录下，复制“`lib/`”目录下的内容到“`/opt/EmbedSky/4.3.3/arm-none-linux-gnueabi/lib/armv4t/lib/`”目录下；复制“`include/`”目录下的内容到“`/opt/EmbedSky/4.3.3/arm-none-linux-gnueabi/lib/usr/include/`”目录下。

推荐从我们的网站下载已经做好的 SDL 的源码，下载地址：http://soft.embedsky.net/files/linux_src/SDL-1.2.13.tar.bz2（在光盘的“Linux 资源\Linux 源码包\apps_20091208.tar.bz2”源码包中已经包含了该源码），解压命令用：“`#tar xvfj SDL-1.2.13.tar.bz2 -C /`”，解压后的源码在“`/opt/EmbedSky/apps/SDL-1.2.13/`”目录下，我们直接使用已经制作好的编译脚本进行配置编译等操作，该编译脚本能完成前面讲到的所有步骤，命令为“`#!/SDL_build`”。

说明：推荐使用我们制作好的脚本编译，那样您可以省去敲入配置命令的麻烦。前面列出来方法是因为以前有客户反馈说不会做程序移植，这里就将详细的步骤罗列出来，希望有所帮助。

Step 4: 编译 mjpg-streamer

我们先进入 mjpg-streamer/目录下，直接使用“`#make`”即可进行编译。编译之后我们会得到：`mjpg-streamer` 和一堆 `input*.so` 或 `output*.so` 的组件，我们在本次试验中使用的有：`input_cmoscamera.so`、`input_uvc.so`、`output_http.so`，对于其他几个组件目前还没有试验其用法，如果您试验出来了，请到论坛发帖告诉我们。

前提：已经完成了前面的准备工作，并且使用的是我们网站提供的已经修改后的源码包；并且这里的 mjpg-streamer 是前面提到的和 uvc-streams 同级的那个目录。

我们将前面提到的*.so 文件复制到开发板的文件系统的“`lib/WebCam/`”目录下（“`WebCam/`”目录是需要自己新建的“`#mkdir WebCam`”），然后复制 mjpg-streamer 文件到开发板的文件系统的“`sbin/`”目录下；然后我们就可以开始测试网络摄像机了。

注意：上面讲到的 `input_cmoscamera.so` 是由我们天嵌科技的 TQ2440 开发板的用户提供的程序编译出来得到的，它的源码存在在我们网站提供的 mjpg-streamer 的源码里面的，对于该部分源码我们没有做任何改动，它里面打开的 CMOS 摄像头（OV9650）的设备名是 `camera`，为此，我们还需要修改我们提供的 Linux-2.6.30.4 的内核的“`drivers/media/video/ov9650.c`”文件的 36 行，将“`video0`”改为“`camera`”，然后重新编译 Linux 内核即可。

Step 5: 编译 uvc-streamer

我们进到 uvc-streamer 目录下，直接使用“`#make`”即可完成编译，编译之后我们会得到 `uvc-stream` 的可执行文件，我们将其复制到开发板的文件系统的“`sbin/`”目录下，然后就可以开始测试网络摄像机了。



Step 6: 测试网络摄像机

测试前提条件:

- 开发板和 PC 直接能使用网络正常通信;
- 开发板的文件系统中已经存在并且安装好了前面提到的组件和可执行文件;
- 开发板使用的 Linux-2.6.30.4 的内核中已经包含了摄像头的驱动(使用我们提供的配置单已经包含了);
- 开发板上已经接好了摄像头, 并且摄像头确保是 OK 的。

说明: 在 <http://bbs.embedsky.net/viewthread.php?tid=630&extra=page%3D1> 这个网址提供的 root_20090811.rar 压缩包中的文件系统里面, 我们为了方便测试, 已经在文件系统的“/sbin/”目录下添加了名为“WebCam_USB”和“WebCam_CMOS”的脚本, 实现 mjpg-streamer 测试 USB 摄像头和 CMOS 摄像头的功能, 默认的 IP 是 192.168.1.6, 省去了输入一长串命令的麻烦。使用方法: “WebCam_USB”或“WebCam_CMOS”即可。

使用 mjpg-streamer 测试 USB 摄像头:

我们在开发板的串口控制台输入命令: `mjpg_streamer -i "/lib/WebCam/input_uvc.so" -o "/lib/WebCam/output_http.so -w 192.168.1.6: 8080"`

注意: 上面的这条命令中-i 中的变量, 可能需要添加-yuv 这个变量, 即变为:

`mjpg_streamer -i "/lib/WebCam/input_uvc.so -yuv" -o "/lib/WebCam/output_http.so -w 192.168.1.6: 8080"`, 然后 192.168.1.6 是指的开发板的 IP, 8080 是打开的端口号。

操作后的截图如下所示: (使用 `ctrl+c` 终止程序, 下面也是如此)



```
Serial-COM1 - SecureCRT
File Edit View Options Transfer Script Tools Help

Serial-COM1
add default gw 192.168.1.2
Done
[11/Aug/2029:15:23:59 +0000] boa: server version Boa/0.94.13
[11/Aug/2029:15:23:59 +0000] boa: server built Jul 29 2009 at 14:27:34.
[11/Aug/2029:15:23:59 +0000] boa: starting server pid=474, port 80
Start Qtopia-2.2.0

Please press Enter to activate this console. Cannot open /dev/mouse0 (No such
file or directory)
Warning: Generating '/opt/Qtopia/etc/dict/words' dawg from word list.
Warning: could not register server

[root@Embedsky /]# mjpg_streamer -i "/lib/webCam/input_uvc.so" -o "/lib/webCa
m/output_http.so -w 192.168.1.6:8080"
MJPEG-streamer [493]: starting application
MJPEG Streamer Version.: 2.0
MJPEG-streamer [493]: MJPG Streamer version.: 2.0

i: using v4L2 device.: /dev/video0
MJPEG-streamer [493]: Using v4L2 device.: /dev/video0

i: Desired Resolution: 640 x 480
MJPEG-streamer [493]: Desired Resolution: 640 x 480

i: Frames Per Second.: 5
MJPEG-streamer [493]: Frames Per Second.: 5

i: Format.....: MJPEG
MJPEG-streamer [493]: Format.....: MJPEG

o: www-folder-path...: 192.168.1.6:8080/
MJPEG-streamer [493]: www-folder-path...: 192.168.1.6:8080/

o: HTTP TCP port.....: 8080
MJPEG-streamer [493]: HTTP TCP port.....: 8080

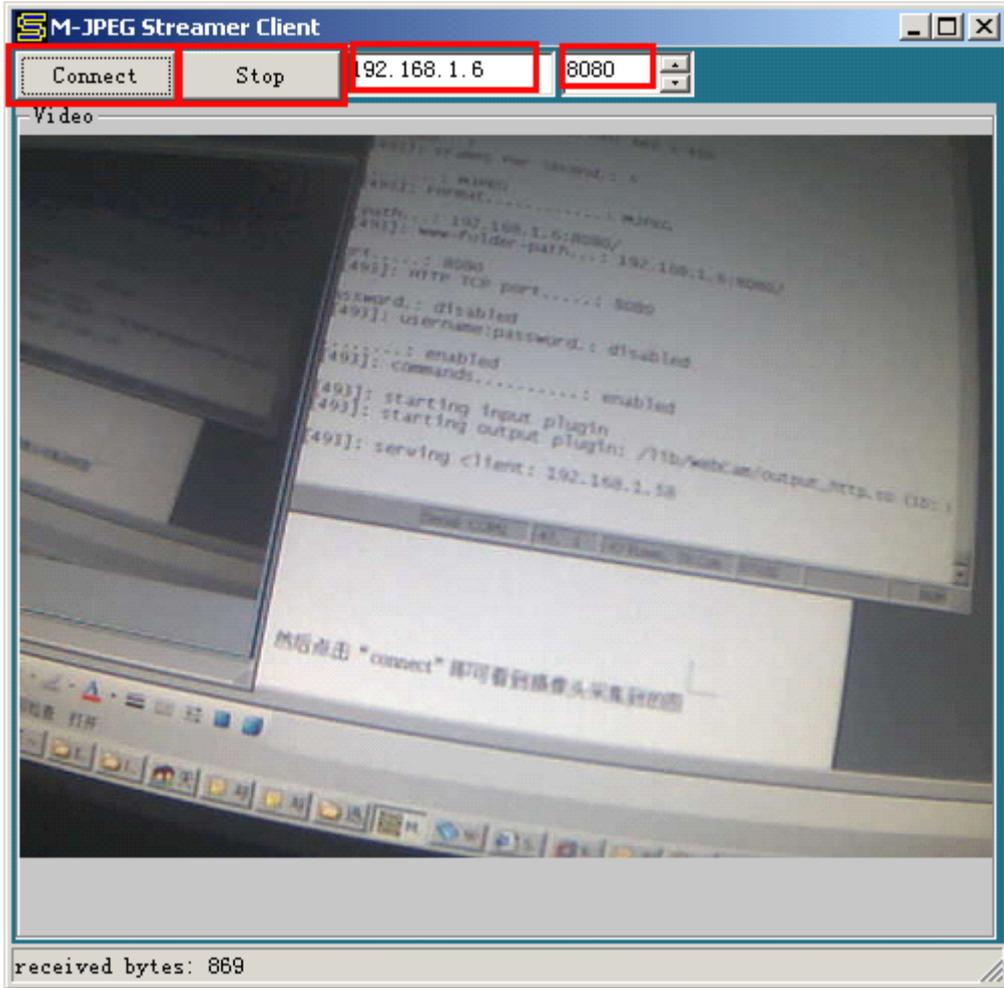
o: username:password.: disabled
MJPEG-streamer [493]: username:password.: disabled

o: commands.....: enabled
MJPEG-streamer [493]: commands.....: enabled

MJPEG-streamer [493]: starting input plugin
MJPEG-streamer [493]: starting output plugin: /lib/webCam/output_http.so (ID: 0)

Ready Serial: COM1 47, 1 47 Rows, 78 Cols VT100 NUM
```

然后我们在 PC 端 (PC 用的是 Windows 平台), 打开前面提到的 viewer.exe, 程序, 然后在 IP 栏目中输入开发板的 IP 和打开的端口号, 然后点击 “connect” 即可看到摄像头采集到的图像。



```
MJPEG-streamer [493]: username:password.: disabled
MJPEG-streamer [493]: o: commands...
MJPEG-streamer [493]: 这行显示的就是PC端打开了接收程序，然后双方完成了通信握手
MJPEG-streamer [493]: starting output plugin: /lib/webCam/output_http.so (ID: 0)
MJPEG-streamer [493]: serving client: 192.168.1.58
```

使用 mjpg-streamer 测试 CMOS 摄像头:

我们在开发板的串口控制台输入命令: `mjpg_streamer -i "/lib/WebCam/input_cmoscamera.so -r 320x256" -o "/lib/WebCam/output_http.so -w 192.168.1.6:8080"`, 操作后的截图如下所示:



```
[root@Embedsky /]# mjpg_streamer -i "/lib/webCam/input_cmoscamera.so -r 320x256"
-o "/lib/webCam/output_http.so -w 192.168.1.6:8080"
MJPEG-streamer [509]: starting application
MJPEG Streamer Version.: 2.0
MJPEG-streamer [509]: MJPG Streamer Version.: 2.0

i: Using CMOS Camera device.: /dev/camera
MJPEG-streamer [509]: Using CMOS Camera device.: /dev/camera

i: Desired Resolution: 320 x 256
MJPEG-streamer [509]: Desired Resolution: 320 x 256

i: Frames Per Second.: 5
MJPEG-streamer [509]: Frames Per Second.: 5

i: Format.....: YUV
MJPEG-streamer [509]: Format.....: YUV

i: JPEG Quality.....: 80
MJPEG-streamer [509]: JPEG Quality.....: 80

o: www-folder-path...: 192.168.1.6:8080/
MJPEG-streamer [509]: www-folder-path...: 192.168.1.6:8080/

o: HTTP TCP port.....: 8080
MJPEG-streamer [509]: HTTP TCP port.....: 8080

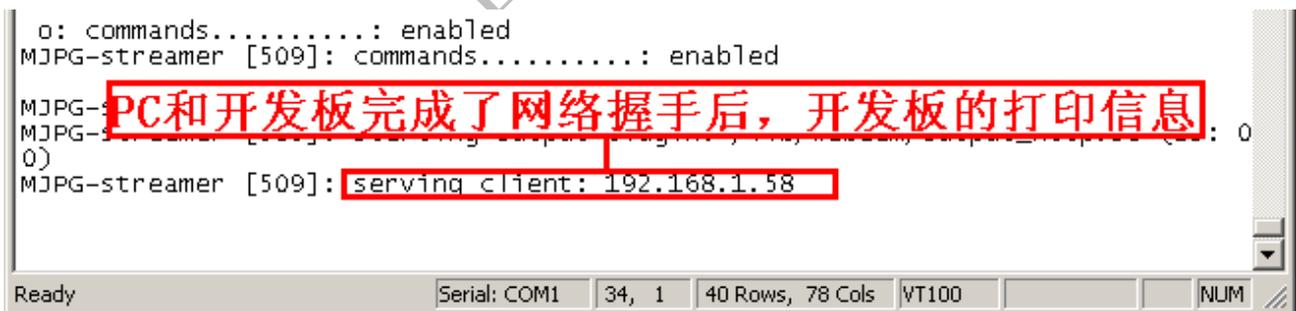
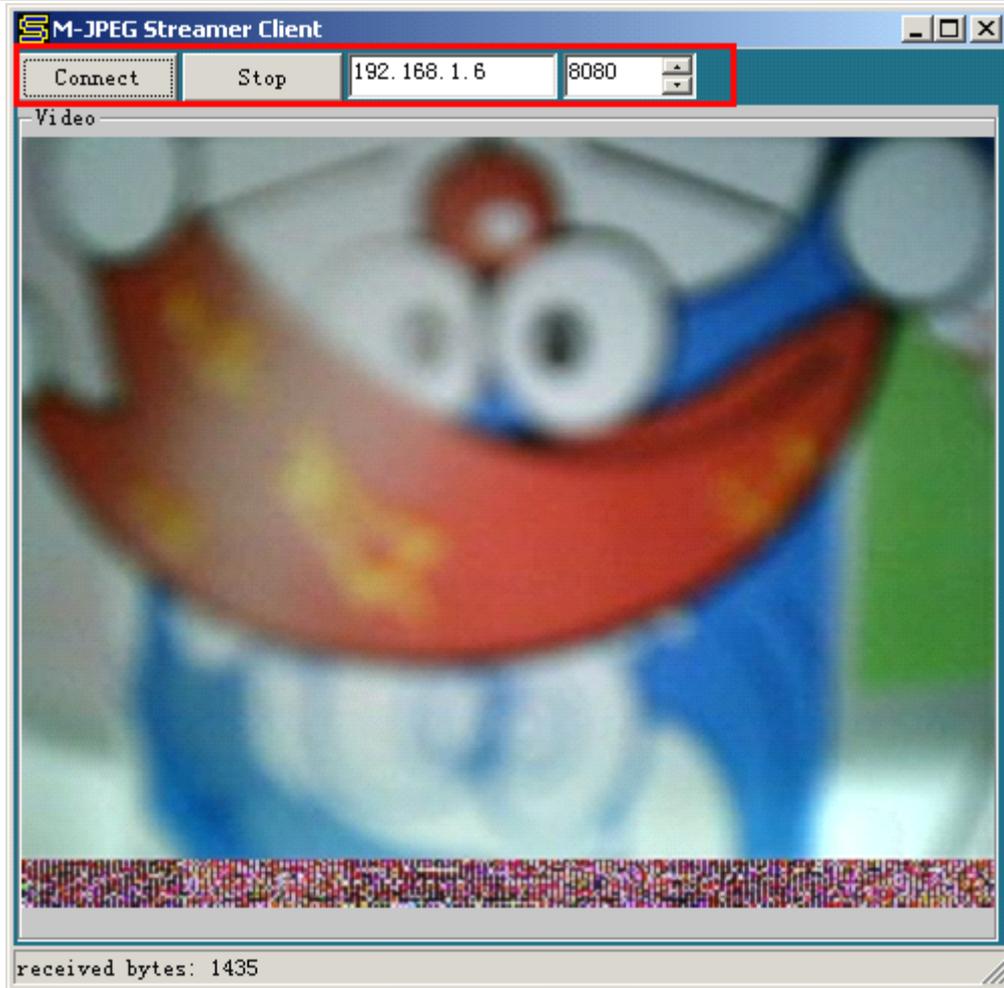
o: username:password.: disabled
MJPEG-streamer [509]: username:password.: disabled

o: commands.....: enabled
MJPEG-streamer [509]: commands.....: enabled

MJPEG-streamer [509]: starting input plugin
MJPEG-streamer [509]: starting output plugin: /lib/webCam/output_http.so (ID: 0)
0)
```

注意：上面命令中的-r 320x252 是指定的采集图片的大小。由于初始化的问题，预览的图片大小除 1280x1024 必须是整数，否则图像会错位。比如 640x512 可以，320x256 也 OK，而 352x288 就会造成图像错位，因为 1280/352 不是一个整数。当然如果不管列错位的话，使用 640x480 也 OK，图像也只是会小一些。

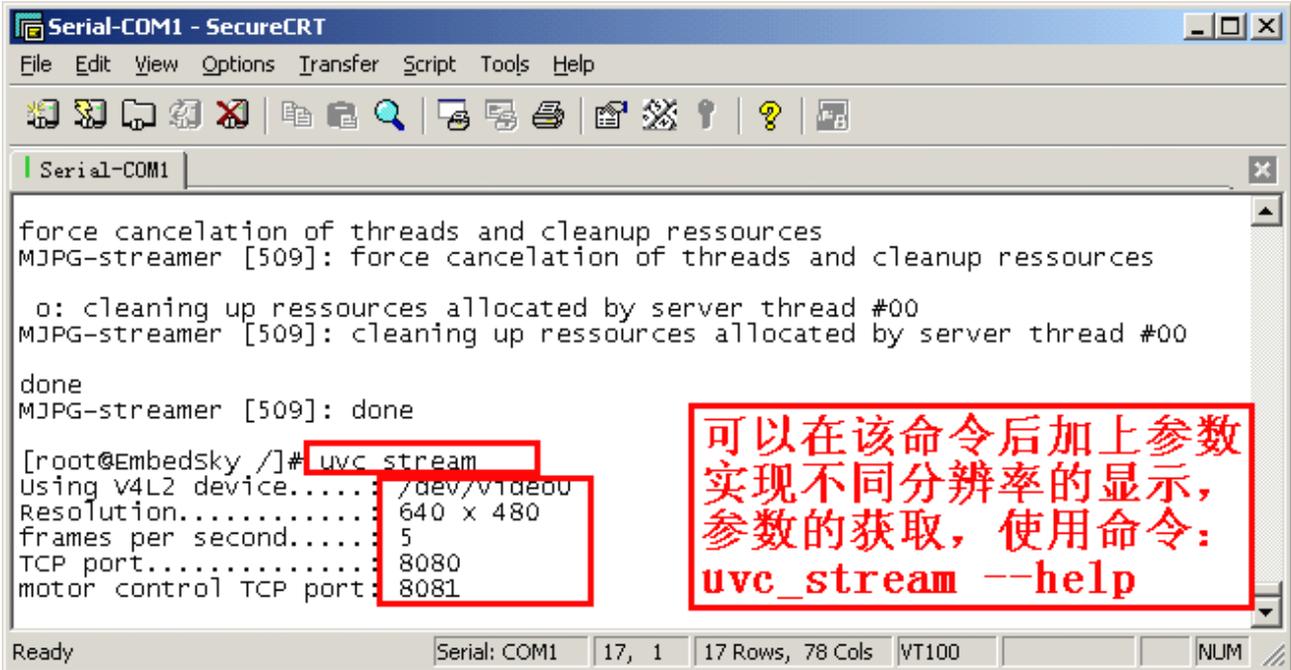
然后我们点击 viewer.exe 程序的 connect 即可看到 CMOS 摄像头采集到的图像了。



注意: Linux-2.6.30.4 内核中我们添加的 CMOS 摄像头 (OV9650) 的驱动中, 将设备名定义为了 video0, 而 input_cmoscamera.so 中定义为 camera, 这里我就需要修改 Linux 内核的 OV9650 的驱动源码的 36 行, 把 video 改为 camera。

使用 uvc-stream 测试 USB 摄像头:

我们直接输入: `uvc-stream`, 然后再打开 viewer.exe, 点击 connect, 我们就可以看到摄像头采集到的图像了, 如下两个截图所示:



注意: CMOS 摄像头不能用它进行测试。

