

Mjpg-streamer 是一个开源软件，用于从 webcam 摄像头采集图像，把它们以流的形式通过基于 ip 的网络传输到浏览器如 Firefox, 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。

mjpg-streamer 可以通过文件或者是 HTTP 方式访问 linux UVC 兼容摄像头。

一、环境

主机环境：ubuntu16.04

目标机：imx6dl-sabresd

交叉工具链：arm-fsl-linux-gnueabi-gcc

二、移植过程

1、配置内核是内核支持 UVC 格式的摄像头（其实默认支持 UVC，它是标准协议），最后一步时根据自己需要进行选择摄像头配置。

Device Drivers--->

<*> Multimedia support--->

[*] Video capture adapters --->

[*] V4L USB devices --->

<*> USB Video Class (UVC)

[*] UVC input events device support

2、重新编译 arm 板的内核，烧录到板子，启动后将 USB 摄像头插进板子串口有信息自动弹出时，即识别成功。

make -j2

3、mjpg-stream 的移植

mjpg-stream 的资料可以在下面这个网址查看（全英文）：

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

mjpg-stream 最新版下载网址是：<http://mjpg-streamer.svn.sourceforge.net/viewvc/mjpg-streamer/>

,

（进去后，点右上的“Download Snapshot”下载最新版的 mjpg-streamer-code-182，因为旧版编译时会出现缺少一个头文件的错误）

mjpg-stream 的移植需要 jpeg 的库，所以我们先移植 jpeg 的库，即需要 libjpeg

第一步：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，下面的配置很重要，自行定义jpeg 安装目录(得先创建那文件夹)，并使用交叉工具链编译jpeg 库（要不然会出现编译 mjpg_streamer 的 input_uvc.o 时找不到兼容的 libjpeg 错误！）

cd jpeg-8b，在命令行分别输入下面命令配置：

```
./configure --prefix=/usr/local/src/out_jpeg -host=arm-none-linux-gnueabi
```

```
./configure CC=arm-linux-gcc --prefix=/usr/local/src/out_jpeg -host=arm-none-linux-gnueabi
```

4)编译

```
make
```

5)安装，执行 **make install** 命令后，在上面配置的/usr/local/src/out_jpeg 文件夹中便出现了一些文件夹，包括 bin、include、lib、share，红色这两个文件夹便是 mjpg_streamer 需要的。

6)拷贝上面的 lib 文件夹即 jpeg 库到文件系统中

```
cp /usr/local/src/out_jpeg/lib /root/rootfs/rootfs/lib -rf
```

(/root/rootfs/rootfs 此目录为自己的文件系统目录，arm 板是挂载 ubuntu 中此目录作为根文件系统的)

第二步、移植 mjpg-streamer

1、将下载最新版本的源码解压至/usr/local/src/，目录 mjpg-streamer-code-182 里内容如下：

```
doc
```

```
mjpg-streamer
```

```
udp_client
```

```
mjpeg-client
```

```
mjpg-streamer-experimental
```

```
uvc-streamer
```

—————部分说明：

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、进入目录

```
cd mjpg-streamer
```

3、（1）修改源码 顶层的 Makefile 及 plugins 文件夹里面所有子层的 Makefile 中的 CC 参数为 arm-linux-gcc，即将所有 “CC = gcc”改为 “CC = arm-linux-gcc”，哈哈，不得不花上几分钟去修改。

（2）进入/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 /usr/local/src/out_jpeg/include
```

修改

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

为

```
(CC)(CFLAGS) -L /usr/local/src/out_jpeg/lib -o
```

```
@inputuvc.cv4l2uvc.lojpegutils.lodyncctrl.lo(LFLAGS)
```

(注: /usr/local/src/out_jpeg/lib 就是上面移植 jpeg 库后安装的目录, 根据自己设置的目录来修改)

这样, mjpg_streamer 就和 jpeg 库绑定了。

4、编译: make (make clean 为清除编译)

在 mjpg-streamer 目录下生成了 input*.so、output*.so 和 mjpg_streamer 可执行文件

5、在开发板挂载的 ubuntu 文件系统目录中 (本人为 /root/rootfs/rootfs) 建立 mjpg-streamer 安装目录

```
mkdir /root/rootfs/rootfs/mjpg-streamer
```

再将源码目录 mjpg-stream 整个拷贝到 /root/rootfs/rootfs/mjpg-streamer 目录下, 然后就可以[测试啦](#)

```
cp mjpg-stream /root/rootfs/rootfs/mjpg-streamer -rf
```

6、测试

(启动 arm 板, 板子应设置好环境变量, 让 arm 板通过 tftp 下载内核和挂载好 ubuntu 中的自制文件系统)

启动后, 插进 USB 摄像头, 进入板子的 mjpg-streamer 目录,

执行 ./start.sh;

若没有出现错误, 则在 PC 机打开一个网页, 输入 <http://ip:8080/?action=stream> 就可以看到图像。

(ip 为 arm 板的 ip)

移植完毕

若出现下面错误:

```
Unable to set format: Invalid argument
```

```
Init v4L2 failed !! exit fatal
```

```
i: init_VideoIn failed
```

错误原因：

市面上大部分摄像头都是支持 YUV 的，而不是 JPEG 的。mjpg-stream 支持 JPEG 和 YUV 两种格式。

解决方法：

```
[fire@fire mjpg-streamer]$ cd mjpg-streamer/plugins/input_uvc/
```

```
[fire@fire input_uvc]$ vi input_uvc.c
```

```
int input_init(input_parameter *param, int id)
{
    char *dev = "/dev/video0", *s;

    int width = 640, height = 480, fps = -1, format = V4L2_PIX_FMT_MJPEG, i;

    if(pthread_mutex_init(&cams[id].controls_mutex, NULL) != 0) {

        IPRINT("could not initialize mutex variable\n");

        exit(EXIT_FAILURE);

    }

    .....
```

将上面红色代码改为：format = V4L2_PIX_FMT_YUYV

以上就设置了默认的图像格式，重新在顶层把模块编译，再次 cp mjpg-stream /root/rootfs/rootfs/mjpg-streamer -rf，arm 板重启后就成功了！