

mjpg-streamer 移植手册

2014.10.30 修订

版权声明

本手册所有权由深圳市微雪电子有限公司独家持有。未经本公司的书面许可，不得以任何方式或形式进行修改、分发或复制本文档的任何部分，否则一切后果由违者自负。

本手册简介

SBC-RK3066 是基于瑞芯微平台所推出的功能扩展板。所有外接模块的驱动程序及应用程序的源码都开源提供，用户可以根据我们提供的程序学习添加更多外设模块。

目前互联网非常强大，很多用户希望能通过网络浏览摄像头，因此我们通过移植 mjpg-stramer，在 SBC-RK3066 实现该功能。mjpg-stramer 是一个开源软件，其基本原理是通过在嵌入式系统中搭建视频流 web 服务器，使得其他用户可以通过网页远程浏览视频图像。

第一章 jpeg 库的移植

1. 获取源码包

把源码/mjpg-streamer 目录下的 jpegsrc.v9a.tar.gz（用户也可以通过网络进行下载：<http://www.iijg.org/files/>）复制到开发板的 ubuntu14.04 系统下，例如复制到 /home/waveshare/mjpg-streamer 目录。

2. 解压源码包

电脑端通过 SSH 连接到开发板，执行以下命令：

```
mkdir /home/waveshare/mjpg-streamer
```

```
cd /home/waveshare/mjpg-streamer
```

解压后产生 jpeg-8b 文件。

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

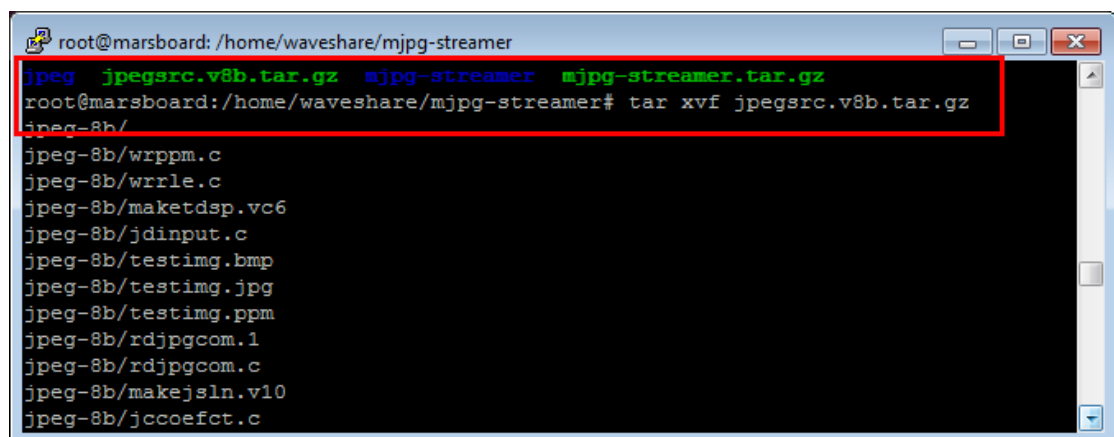
3. 配置源码包

创建文件夹用于存放生成的 jpeg 依赖库文件

```
mkdir /home/waveshare/mjpg-streamer/jpeg
```

```
cd jpeg-8b
```

```
./configure --prefix=/home/waveshare/API/mjpg-streamer/jpeg --host=
```



```
root@marsboard: /home/waveshare/mjpg-streamer
jpeg jpegsrc.v8b.tar.gz mjpg-streamer mjpg-streamer.tar.gz
root@marsboard:/home/waveshare/mjpg-streamer# tar xvf jpegsrc.v8b.tar.gz
jpeg-8b/
jpeg-8b/wrppm.c
jpeg-8b/wrrle.c
jpeg-8b/maketsdsp.vc6
jpeg-8b/jdinput.c
jpeg-8b/testing.bmp
jpeg-8b/testing.jpg
jpeg-8b/testing.ppm
jpeg-8b/rdjpgcom.1
jpeg-8b/rdjpgcom.c
jpeg-8b/makejs1n.v10
jpeg-8b/jccofct.c
```

参数说明：

`./configure`：配置源代码树

`--prefix=XX`：安装目录

`--host`：指定编译器

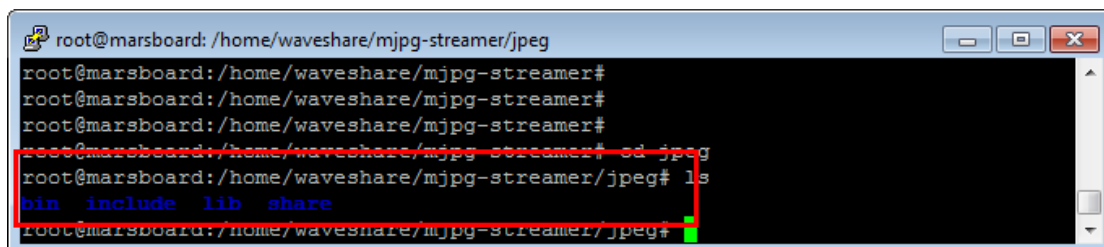
4. 编译安装

执行以下命令：

```
make
```

```
make install
```

/home/waveshare/mjpg-streamer/jpeg 路径下会生成 bin, include, lib, share 目录。



```
root@marsboard: /home/waveshare/mjpg-streamer/jpeg
root@marsboard: /home/waveshare/mjpg-streamer#
root@marsboard: /home/waveshare/mjpg-streamer#
root@marsboard: /home/waveshare/mjpg-streamer# cd jpeg
root@marsboard: /home/waveshare/mjpg-streamer/jpeg# ls
bin  include  lib  share
root@marsboard: /home/waveshare/mjpg-streamer/jpeg#
```

第二章 mjpg-streamer 的移植

1. 获取源码包

把源码/mjpg-streamer 目录下的 **mjpg-streamer.tar.gz**（用户也可以通过网络进行下载：<http://sourceforge.net/projects/mjpg-streamer/>）复制到开发板的 ubuntu 系统下，例如复制到/home/waveshare/ mjpg-streamer 目录。

2. 解压源码包

终端定位到上述目录下，执行以下命令解压：

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

解压后产生 mjpg-streamer 文件。

3. 配置源码包

- 1) 打开 mjpg-streamer 文件夹：

```
cd /home/waveshare/mjpg-streamer/mjpg-streamer
```

- 2) 修改 plugins/input_uvc/目录下的 Makefile:

```
vim plugins/input_uvc/Makefile
```

- 3) 指定 mjpg 图形库路径（路径必须和第一章所设相同）：

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

后面加入图形库链接路径：

```
-I/home/waveshare/mjpg-streamer/jpeg/include
```

- 4) 在\$(CC) \$(CFLAGS) -ljpeg -o \$@ input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo “-o” 前加入 jpeg 图形库的 lib 路径:

-L/home/waveshare/mjpg-streamer/jpeg/lib

如下图所示:

```

1 #####
2 #
3 # Purpose: Makefile for "M-JPEG Streamer"
4 # Author.: Tom Stoeveken (TST)
5 # Version: 0.3
6 # License: GPL
7 #
8 #####
9
10 CC = gcc
11
12 OTHER_HEADERS = ../../mjpg_streamer.h ../../utils.h ../output.h ../input.h
13
14 CFLAGS += -O1 -D_LINUX -D_GNU_SOURCE -Wall -shared -fPIC -I/home/waveshare/mjpg-s
15 treamer/jpeg/include
16
17 #CFLAGS += -g
18 #CFLAGS += -DDEBUG
19
20 ifeq ($(USE_LIBV4L2),true)
21 LFLAGS += -lv4l2
22 CFLAGS += -DUSE_LIBV4L2
23 endif
24
25 LFLAGS += -ljpeg
26
27 all: input_uvc.so
28
29 clean:
30 rm -f *.a *.o core *~ *.so *.lo
31
32 input_uvc.so: $(OTHER_HEADERS) input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo
33 $(CC) $(CFLAGS) $(LFLAGS) -L/home/waveshare/mjpg-streamer/jpeg/lib -o $@
34 input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo
35
36 v4l2uvc.lo: huffman.h uvc_compat.h v4l2uvc.c v4l2uvc.h
37 $(CC) -c $(CFLAGS) -o $@ v4l2uvc.c
38
39 jpeg_utils.lo: jpeg_utils.c jpeg_utils.h
40 $(CC) -c $(CFLAGS) -o $@ jpeg_utils.c
41
42 dynctrl.lo: dynctrl.c dynctrl.h
43 $(CC) -c $(CFLAGS) -o $@ dynctrl.c
44
45 :set nu
46
47 42,35-42 All

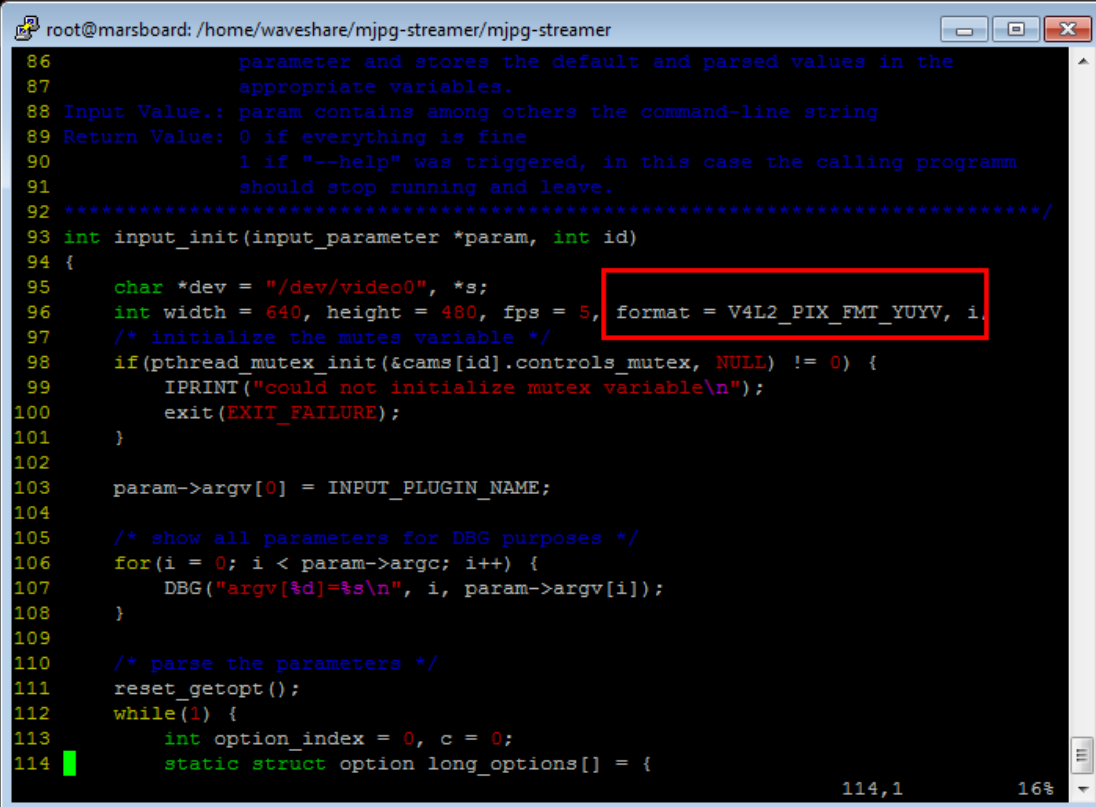
```

4. 修改视频的编码格式

vim plugins/input_uvc/input_uvc.c

将 format = V4L2_PIX_FMT_MJPEG 修改为 format = V4L2_PIX_FMT_YUYV

如下图所示:



```

86         parameter and stores the default and parsed values in the
87         appropriate variables.
88 Input Value.: param contains among others the command-line string
89 Return Value: 0 if everything is fine
90               1 if "--help" was triggered, in this case the calling program
91               should stop running and leave.
92 *****/
93 int input_init(input_parameter *param, int id)
94 {
95     char *dev = "/dev/video0", *s;
96     int width = 640, height = 480, fps = 5, format = V4L2_PIX_FMT_YUYV, i;
97     /* initialize the mutex variable */
98     if(pthread_mutex_init(&cams[id].controls_mutex, NULL) != 0) {
99         IPRINT("could not initialize mutex variable\n");
100         exit(EXIT_FAILURE);
101     }
102
103     param->argv[0] = INPUT_PLUGIN_NAME;
104
105     /* show all parameters for DBG purposes */
106     for(i = 0; i < param->argc; i++) {
107         DBG("argv[%d]=%s\n", i, param->argv[i]);
108     }
109
110     /* parse the parameters */
111     reset_getopt();
112     while(1) {
113         int option_index = 0, c = 0;
114         static struct option long_options[] = {

```

5. 编译

```
make
```

将在本目录下生成视频流服务器运行时需要的动态库及相关脚本。

6. 更新所需的库文件

```
sudo apt-get install subversion
```

```
sudo apt-get install libv4l-dev
```

```
sudo apt-get install libjpeg62-dev
```

```
sudo apt-get install imagemagick
```