

dddawant

于 2012-12-31 03:56:24 发布

8262

☆ 收藏 6

原版

分类专栏:

Linux

X86

Linux

同时被 2 个专栏收录 *

0 订阅

2 篇文章

订阅专栏

v4l (video for Linux) 使用摄像头实验的一点小感慨

前言

学习Linux快1个月了，Linux视频教程看完了，然后在虚拟机上装了个CentOS6.3把该做的实验大多也做了，突然就想玩玩刚买来的星瀚38摄像头（型号K2200），我自己定了个计划，先写个摄像头驱动再写个应用层上面与图像有关的能使什么程序。

正文

1，先是驱动，我在想38摄像头的该USB总线协议部分Linux应该已经有了，我只需要于这个USB总线协议上增加一层摄像头的通信协议就可以了，于是我就想把我的摄像头拆了看看里面用的是什么东西，结果没有余得拆，于是我只能去淘宝网站物品信息部分看看有没有我的K2200所用芯片信息，结果查由于芯片供应商是中星微但具体芯片型号没有，我余不得拆，怎么办，我突然想起我的摄像头是免驱的，既然windows都自带它的驱动，就说明芯片那边使用的协议一定是符合某个标准的，上于是在网上捣鼓了一下，我靠，有好多论文和论坛都在讲关于USB摄像头驱动，居然有这么多人在玩摄像头，看来我落伍了，看看看看我渐渐的明白了，原来有个V4L (video for Linux) 协议，果不其然，既然Linux都已经自带了摄像头驱动，那我就不了写了，但是我要看看我的CentOS6.3有没有集成摄像头驱动，我把摄像头插上，windows给我自动安装程序，安装完了，好像又安装了一个'mmware usb device'什么东西，不管它（后来才知道windows无头后再启动虚拟机就不安装'mmware usb device'了，要设置下：虚拟机->可移动设备->usb设备上钩一下），然后再lsmod | grep video下就能看到kvcvvideo，videodev之类的模块如图1，用ll /dev命令也能看到/video0这个设备文件了如图2。

```
File Edit View Search Terminal Help
[root@CentOS camera]# lsmod | grep video
kvcvvideo          55168  0
videodev           65378  1 kvcvvideo
i2c_core           25825  2 videodev,i2c_piix4
[root@CentOS camera]#
```

图1

```
crw-rw---- 1 vcsa tty      7, 132 Dec 30 22:08 vcsa0
crw-rw---- 1 vcsa tty      7, 133 Dec 30 22:08 vcsa1
crw-rw---- 1 vcsa tty      7, 134 Dec 30 22:08 vcsa6
crw-rw---- 1 root root     10,  63 Dec 30 22:07 vga_arbiter
crw-rw---- 1 root video     81,   0 Dec 30 22:07 videom
crw-rw-rw- 1 root root      1,   5 Dec 30 22:07 video0
[root@CentOS camera]#
```

图2

2，应用层，上网搜一搜，无聊，也有好多程序，很给力，看来我有了很好的学习材料，先给由一些传送门
http://blog.csdn.net/id_south/article/details/4288836（这位叫id_south的朋友总结了网上的USB摄像头程序然后给出了自己的想法和源程序，我就是借鉴的这个，稍微改一下能通过gcc编译），还有人把网上的程序搜集到一块了，比如<http://blog.chinaunix.net/uid-27875-id-3069828.html>，好多好多，我不一一列举了。我把id_south的程序一个字母一个字母的用gedit编译器敲进去，形成了如下的程序。

v4l.h头文件

```
1  #ifndef _V4L_H_
2  #define _V4L_H_
3
4  #include <stdio.h>
5  #include <stdlib.h> //stdio.h and stdlib.h are needed by perror function
6  #include <sys/stat.h>
7  #include <sys/types.h>
8  #include <fcntl.h> //O_RDONLY
9  #include <unistd.h>
10 #include <sys/mman.h> //unistd.h and sys/mman.h are needed by mmap function
11 #include <stdbool.h> //false and true
12 #include <sys/ioctl.h>
13 #include <linux/videodev.h> //v4l API
14
15 typedef struct _v4l_struct
16 {
17     int fd;
18     struct video_capability capability;
19     struct video_picture picture;
20     struct video_mmap mmap;
21     struct video_mbuf mbuf;
22     unsigned char *map;
23     int frame_current;
24     int frame_using[VIDEO_MAX_FRAME];
25 }v4l_device;
26
27 extern int v4l_open(char *dev,v4l_device *vd);
28 extern int v4l_close(v4l_device *);
29 extern int v4l_get_capability(v4l_device *);
30 extern int v4l_get_picture(v4l_device *);
31 extern int v4l_get_mbuf(v4l_device *);
32 extern int v4l_set_picture(v4l_device *,int,int,int,int);
33 extern int v4l_grab_picture(v4l_device *,unsigned int);
34 extern int v4l_mmap_init(v4l_device *);
35 extern int v4l_grab_init(v4l_device *,int,int);
36 extern int v4l_grab_frame(v4l_device *,int);
37 extern int v4l_grab_sync(v4l_device *);
38
39 #define DEFAULT_DEVICE "/dev/video0"
40
41 int v4l_open(char *dev,v4l_device *vd)
42 {
43     if(!dev)
44         dev=DEFAULT_DEVICE;
45     if((vd->fd=open(dev,O_RDONLY))<0)
46     {
47         perror("v4l_open fail");
48         return -1;
49     }
50     if(v4l_get_capability(vd))
51         return -1;
52     printf("video capture device name:%s\n",vd->capability.name);
53     if(v4l_get_picture(vd))
54         return -1;
55     printf("frames number is %d\n",vd->mbuf.frames);
56     return 0;
57 }
58
59
60 int v4l_close(v4l_device *vd)
61 {
62     munmap(vd->map,vd->mbuf.size);
63     close(vd->fd);
64     return 0;
65 }
66
67 int v4l_get_capability(v4l_device *vd)
68 {
69     if(ioctl(vd->fd,VIDIOCGCAP,&vd->capability)<0)
70     {
71         perror("v4l_get_capability fail");
72         return -1;
73     }
74     return 0;
75 }
76
77 int v4l_get_picture(v4l_device *vd)
78 {
79     if(ioctl(vd->fd,VIDIOCGPICT,&vd->picture)<0)
80     {
81         perror("v4l_get_picture fail");
82         return -1;
83     }
84     return 0;
85 }
86
87 int v4l_get_mbuf(v4l_device *vd)
88 {
89     if(ioctl(vd->fd,VIDIOCGMBUF,&vd->mbuf)<0)
90     {
91         perror("v4l_get_mbuf fail");
92         return -1;
93     }
94     return 0;
95 }
96
97 int v4l_set_picture(v4l_device *vd,int br,int hue,int col,int cont,int white)
98 {
99     if(br)
100         vd->picture.brightness=br;
101     if(hue)
102         vd->picture.hue=hue;
103     if(col)
```

```
184     vd->picture.colour=col;
185     if(cont)
186         vd->picture.contrast=cont;
187     if(white)
188         vd->picture.whiteness=white;
189     if(ioctl(vd->fd,VIDIOCSPIC, &vd->picture)<0)
190     {
191         p e r r o r ("v4l_set_picture fail");
192         return -1;
193     }
194     return 0;
195 }
196
197 int v4l_grab_picture(v4l_device *vd,unsigned int size)
198 {
199     if(read(vd->fd,vd->map,size)==0)
200         return -1;
201     return 0;
202 }
203
204 int v4l_mmap_init(v4l_device *vd)
205 {
206     if(v4l_get_sbuf(vd)<0)
207         return -1;
208     if((vd->map=(unsigned char*)mmap(0,vd->sbuf.size,PROT_READ|PROT_WRITE,MAP_SHARED,vd->fd,0))<0)
209     {
210         p e r r o r ("v4l_mmap_init fail");
211         return -1;
212     }
213     return 0;
214 }
215
216 int v4l_grab_init(v4l_device *vd,int width,int height)
217 {
218     vd->map.width=width;
219     vd->map.height=height;
220     vd->map.format=vd->picture.palette;
221     vd->frame_current=0;
222     vd->frame_using[0]=false;
223     vd->frame_using[1]=false;
224     return v4l_grab_frame(vd,0);
225 }
226
227 int v4l_grab_frame(v4l_device *vd,int frame)
228 {
229     if(vd->frame_using[frame])
230     {
231         f p r i n t f ( s t d e r r , "v4l_grab_frame %d is already used\n", frame);
232         return -1;
233     }
234     vd->map.frame=frame;
235     if(ioctl(vd->fd,VIDIOCMCAPTURE, &vd->map)<0)
236     {
237         p e r r o r ("v4l_grab_frame fail");
238         return -1;
239     }
240     vd->frame_using[frame]=true;
241     vd->frame_current=frame;
242     return 0;
243 }
244
245 int v4l_grab_sync(v4l_device *vd)
246 {
247     if(ioctl(vd->fd,VIDIOCSYNC, &vd->frame_current)<0)
248     {
249         p e r r o r ("v4l_grab_sync fail");
250         return -1;
251     }
252     vd->frame_using[vd->frame_current]=false;
253     return 0;
254 }
255
256 #endif
257
```

app-camera.c测试程序

```
1 #include "v4l.h"
2
3 v4l_device vd;
4
5 void main()
6 {
7     v4l_open(DEFAULT_DEVICE, &vd);
8     v4l_mmap_init(&vd);
9     v4l_grab_init(&vd,320,240);
10    v4l_grab_sync(&vd);
11
12    /*while(1)
13    {
14        vd->frame_current+=1;
15        v4l_grab_frame(&vd,vd->frame_current);
16        v4l_grab_sync(&vd);
17    } */
18    v4l_close(&vd);
19 }
20
```

编译运行结果出乎意料产生了v4l_get_capability fail: Invalid argument或者VIDIOCGCAP:Invalid argument, 只要是使用v4l给的ioctl命令都是这样的错误。如图3所示

```
[root@centos camera]# ./app-camera
v4l_get_capability fail: Invalid argument
v4l_get_sbuf fail: Invalid argument
v4l_grab_frame fail: Invalid argument
v4l_grab_sync fail: Invalid argument
[root@centos camera]#
```

图3

我开始百度了, 关键词是VIDIOCGCAP : invalid argument, 结果如图4



图4

我想说的是, 我搜了半天, 遇到很多跟我同样问题的人, 但都没有得到给定的结果, 我真的很崩溃。其实我不怎么用google, google的主页不知什么原因老是打不开, 而且基本上一搜都是英文, 我不喜欢英文。那么就放弃.....没办法, 我只能去google, 心能没办法。

我开始google了, 关键词同样是VIDIOCGCAP : invalid argument, 结果如图5

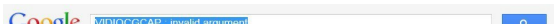




图5

我想说的是第二个就出现了我要的答案。这个老外是这样讲的：I'm having some issue working with V4L (the API that provides unified access to various video capturing for Linux). I'm trying to make a VIDIOCGCAP ioctl call, but I get an INVALID ARGUMENT error. 并贴上了自己的测试结果和源码，如图6。

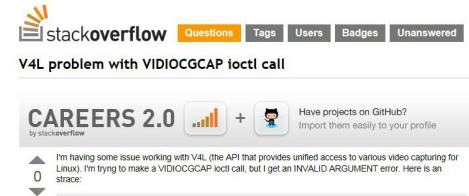


图6

然后当真，就有人回复Perhaps you need to try using V4L2 instead, namely the VIDIOC_QUERYCAP ioctl.大致意思是你使用v4l2试一试，命令宏都差不多，如图7。



图7

ok,it works...我表示和这个老外一样的惊讶。

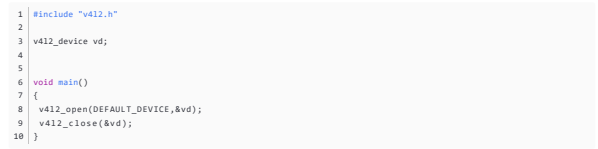
跟帖更精彩，例如，FYI, V4L is an antiquated interface and most new devices are brought-up with V4L2 support.

我验证了一下这个解决方法。

把v4l.h头文件修改为v4l2.h, v4l2.h如下



把测试程序app-camera.c改为test.c, test.c如下



运行结果如图8所示。

```
[root@CentOS camera]# ./test
video capture device name:uvcvideo
[root@CentOS camera]#
```

圖8

我得到了我想要的结果。

后记

改天写

✎ 文章知识点与官方知识档案匹配，可进一步学习相关知识

CS入门技能树 > Linux环境安装 > 安装CentOS 1973 人正在系统学习中

ddsdswent
粉丝 10 篇 · 暂无认证

5	37万+	177万+	4万+	
原创	周排名	总排名	访问	等级
434	136	16	14	
积分	粉丝	获赞	评论	收藏

私信
关注

[搜博主文章](#)

热门文章

[用v4l2和rimebuffer实现usb摄像头视频采集并显示](#) @ 17755

[突然想玩玩ucgui](#) @ 11112

[v4l2video for Linux使用摄像头实验的一点小心得](#) @ 8261

[android模拟连接PC](#) @ 3483

[用户进程与windows操作系统回收资源](#) @ 1222

最新文章

[用v4l2和rimebuffer实现usb摄像头视频...
ackey_L_rgh 的顺序不对导致的](#)

[win97for Linux使用摄像头实验的“...
w4970651：非常感谢](#)

[用v4l2和rimebuffer实现usb摄像头视频...
龙图楼：赞，支持原创！](#)

[用v4l2和rimebuffer实现usb摄像头视频...
qq_1342921：你可以参考下述内容：\[http://blog.csdn.net/stmang/article/details=...
色色色：yuy+特RtG565两个字，RTG565是缩写，一个像素占两个字，其中，\]\(#\)](#)

您愿意向朋友推荐“博客园”吗？






强烈不推荐
不推荐
一般般
推荐
强烈推荐

最新文章

[android模拟连接PC](#)

[突然想玩玩ucgui](#)

[用v4l2和rimebuffer实现usb摄像头视频采集并显示](#)

2014年 1篇
2013年 2篇

2012年 2篇

v4l2获取视频yuv视频文件
linux c语言编写的**V4L2**视频编码文件，代码支持yuv yuv420

open VIDEOIO(/v4l2/dev/video0): can't open camera by index LikeShadow 1729

在 linux 下运行 opencv 使用摄像头采集图像时，弹出这个错误：网上各种找，各种试，都不行，最后还是请教大佬，说是修改下 USB 的兼容性，修改为 ...

评论 2 您还未登录，请先 [登录](#) 后发表或查看评论

由于本人是菜鸟一个,所以是需要上网搜一搜看大家是如何做的,当然开始都是理不出一个头绪,但是很多文章都提到了[video4linux\(v4l\)](#),所以我觉得工作的...

video4linux(v4l)使用摄像头的实例基础教程与体会(转) 5-3

video4linux头文件可以在/usr/include/linux下找到,里面包含了对v4l各种结构的定义,以及各种ioctl的使用方法,所以在下文中有有关v4l的相关结构体并不做详细

由于cmos摄像头模组或者usb摄像头采集的数据为yuy2格式的视频数据，所以需要安装一个yuy2的播放器播放这种格式的视频数据

video4linux(v4l)使用摄像头的实例基础教程与体会(二) shang_125的专栏 · 1218

前面已经说过使用v4l视频编程的流程和对文件操作并没有什么本质的不同,大量的流程如下:

1. 打开视频设备(通常是/dev/video0)
2. 获得设备信...

video4linux(v4l)使用摄像头的实例基础教程与体会(四) 3-16

获得图像前的初始化工作v4l grab_init()该函数十分简单直接贴上去,其中将_vd->frame_using[0]和vd->frame_using[1]都设为FALSE表示两帧的截取都...

Linux - 使用V4L2 (总结)

linux的v4l2驱动, V4L2 Linux 驱动简介(1) weixin_39783857 的博客 66

笔者最近有机会写了一个摄像头的驱动, 是“One laptop per child”项目中的摄像头专用的, 这个驱动使用了为此目的而设计的内核API: 的 Video4Linux2

Android设备驱动之——V4L2Video for Linux TwoV4L2的是V4L的第二个版本。原来的V4L被引入到Linux内核2.1.x的开发周期后期。Video4Linux2修正了...

Q1--Linux平台QI5用V4L2读取摄像头, 实现实时画面和拍照 sunny_hu92的博客 · 1万

平台: ubuntu14.04 工具: qt5.7 opensource版本 Desktop 1. V4L2的介绍: 摘自<http://www.cnblogs.com/silence-hust/p/4464291.html> 大部分都是一样的...

V4L2库使用

JIANJIANJIAN1111的博文 © 453

v4l2，一开始听到这个名字的时候，以为又是一个很难很难的模块，涉及到视频的处理，后来在网上各种找资料后，才发现其实v4l2已经封装好了驱...

Linux V4L2 人工智能、深度学习、人脸识别 © 2225

V4L是 Video for Linux 的缩写，它是 Linux 内核中关于视频设备的子系统，它为 Linux 下的视频驱动提供了统一的接口，使得应用程序可以使用统一的 API 调...

Linux下用V4L2采集摄像头图像和视频 Wind blog 5168

V4L2, 即 video for linux 2, V4L的第二版, linux下的视频库, 非常方便用于采集摄像头数据, 使用方法一般有以下流程 打开设备——(检查设备属性)——

Linux一个摄像头2个video,V4L2视频开发 (摄像头) weicon_33644009的博文 © 240

video4linux(v4l)使用摄像头的实例基础教程与体会(五) shang_125的专栏 · 100%

2 上述v4l库使用的方法给出了上述的一些代码，这里用一些简单的代码表明如何来使用它。上文中已经说过将相关结构体和函数的定义放到一个名为v4l.h。

linux 里的v4l函数,video for linux——v4l v4l (Video for Linux 标准) weixin_30416973 的博客 6 个月前

v4l是Linux中提供的一个音视频接口规范,所有的音视频设备的驱动编写要用到的这些接口Video4Linux其中用到的数据结构有: 2-28, I was sitting just by th...

我们一起学linux之V4L2摄像头应用流程 [最新发布](#) [转载和创作优秀的博客](#) [146](#)

嵌入式linux中摄像头使用,video4linux(v4l)使用摄像头的实例基础教程与体会(顶) weixin_32252271 的博客 5

1.video4linux基础相关:1.v4l的介绍与一些基础知识:的介绍,首先说明一下video4linux(v4l),它是一些视频系统,视频软件,音频软件的基础,经常使用在

基于Video4Linux的视频采集模块开发 热门推荐 复兴之路 1万+

linux系统中，摄像头驱动程序安装好后，为了进行视频采集必须添加Video4Linux模块，从而可以通过Video4Linux模块提供的编程接口(API)从摄像头设备...

基于linux视频架构V4L2分析——video for linux (二) laoyihui 1478

V4L2跟V4L有较大的改动，已经称为3.0的接口标准，涵盖了大多数的视频或其它dvd和FM等驱动，较向V4L2靠拢，V4L2采用流水线的工作方式，基本遵

video4linux(v4l)使用摄像头的实例基础教程与体会 复兴之路 404

1. video4linux 基础相关 1.1 v4l的介绍与一些基础知识的介绍 1.首先说明一下video4linux(v4l), 它是一些视频系统、视频软件、音频软件的基础, ...

“相关推荐”对你有帮助?

©2022 CSDN 皮肤主题: 大白 设计师: CSDN官方博客 返回首页

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

中国互联网举报中心 Chrome商店下载 ©1999-2022北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照