


原创 xukai871105 2014-08-09 15:22:09 12310 收藏 6 版权

分类专栏： 树莓派 文章标签： 树莓派 sysfs GPIO

 树莓派 专栏收录该内容

29 订阅 37 篇文章 订阅专栏

## 0 前言

本文描述如果通过文件IO sysfs方式控制树莓派 GPIO端口。通过sysfs方式控制GPIO，先访问/sys/class/gpio目录，向export文件写入GPIO编号，使得该GPIO的操作接口从内核空间暴露到用户空间，GPIO的操作接口包括direction和value等，direction控制GPIO方向，而value可控制GPIO输出或获得GPIO输入。

Linux学习可从应用出发，先不纠结Linux驱动编写，先把Linux给玩起来。

### 【相同与不同】

本文和【[EasyARM i.mx28学习笔记——文件IO方式操作GPIO](#)】内容相似，大部分代码相同。通过文件IO操作可以有效地避免平台差异，虽然EasyARM im287平台和树莓派完全不同，但是通过sysfs操作GPIO实现代码大致相同。

和EasyARM im287不同，此处并没有使用交叉编译工具，有树莓派中的gcc工具链编译链接获得可执行文件，而EasyARM im287并不能这样操作。EasyARM im287采用busybox指令集，这也与树莓派中的debian指令集存在差异。

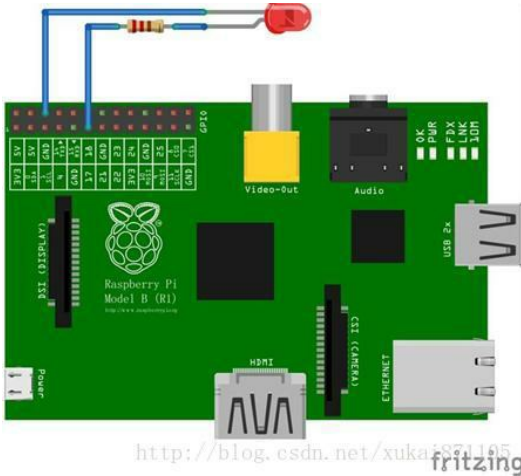
### 【相关博文】

- 【[EasyARM i.mx28学习笔记——文件IO方式操作GPIO](#)】
- 【[树莓派学习笔记——Shell脚本操作GPIO](#)】
- 【[树莓派学习笔记——交叉编译工具链](#)】
- 【[Linux学习笔记——例说makefile 索引博文](#)】

### 【代码仓库】

代码仓库位于bitbucket——[rpi-gpio-sysfs](#)，请使用Hg克隆或者直接下载zip包。请不要使用任何版本的IE浏览器访问链接，除非你已经知道所使用的IE浏览器符合HTML5标准。推荐使用谷歌或者火狐浏览器访问，若使用国产双核浏览器请切换到极速模式。

### 【原理图示意】



<http://blog.csdn.net/xukai871105>

图1 连线示意图

## 1 暴露GPIO操作接口

### 分类专栏

|   |            |     |
|---|------------|-----|
|  | Kubernetes | 1篇  |
|  | NetCore    | 1篇  |
|  | 其他         | 20篇 |
|  | 树莓派        | 37篇 |
|  | 物联网学习笔记    | 61篇 |
|  | Linux学习笔记  | 22篇 |
|  | python     | 6篇  |
|  | 嵌入式ARM     | 25篇 |
|  | 单片机        | 28篇 |
|  | 个人计划/个人随笔  | 4篇  |
|  | Java与后端开发  | 2篇  |

```

1 static int gpio_export(int pin)
2 {
3     char buffer[BUFFER_MAX];
4     int len;
5     int fd;
6
7     fd = open("/sys/class/gpio/export", O_WRONLY);
8     if (fd < 0) {
9         fprintf(stderr, "Failed to open export for writing!\n");
10        return(-1);
11    }
12
13    len = snprintf(buffer, BUFFER_MAX, "%d", pin);
14    if (write(fd, buffer, len) < 0) {
15        fprintf(stderr, "Fail to export gpio!");
16        return -1;
17    }
18
19    close(fd);
20    return 0;
21 }

```

## 2 隱藏GPIO操作接口

```

1 static int gpio_unexport(int pin)
2 {
3     char buffer[BUFFER_MAX];
4     int len;
5     int fd;
6
7     fd = open("/sys/class/gpio/unexport", O_WRONLY);
8     if (fd < 0) {
9         fprintf(stderr, "Failed to open unexport for writing!\n");
10        return -1;
11    }
12
13    len = snprintf(buffer, BUFFER_MAX, "%d", pin);
14    if (write(fd, buffer, len) < 0) {
15        fprintf(stderr, "Fail to unexport gpio!");
16        return -1;
17    }
18
19    close(fd);
20    return 0;
21 }

```

## 3 配置GPIO方向

```

1 static int gpio_direction(int pin, int dir)
2 {
3     static const char dir_str[] = "in\0out";
4     char path[DIRECTION_MAX];
5     int fd;
6
7     snprintf(path, DIRECTION_MAX, "/sys/class/gpio/gpio%d/direction", pin);
8     fd = open(path, O_WRONLY);

```

```

9     if (fd < 0) {
10         fprintf(stderr, "failed to open gpio direction for writing!\n");
11         return -1;
12     }
13
14     if (write(fd, &dir_str[dir == IN ? 0 : 3], dir == IN ? 2 : 3) < 0) {
15         fprintf(stderr, "failed to set direction!\n");
16         return -1;
17     }
18
19     close(fd);
20     return 0;
21 }

```

【简单说明】

【1】dir\_str[dir == IN ? 0 : 3], dir == IN ? 2 : 3 如果输入为常数宏IN, 取dir\_str[0]="in";若输入常数宏为OUT, 取dir\_str[0]="out"。此处巧妙的使用了在数组中的"\0"。

## 4 控制GPIO输出

```

1 static int gpio_write(int pin, int value)
2 {
3     static const char values_str[] = "01";
4     char path[DIRECTION_MAX];
5     int fd;
6
7     snprintf(path, DIRECTION_MAX, "/sys/class/gpio/gpio%d/value", pin);
8     fd = open(path, O_WRONLY);
9     if (fd < 0) {
10         fprintf(stderr, "failed to open gpio value for writing!\n");
11         return -1;
12     }
13
14     if (write(fd, &values_str[value == LOW ? 0 : 1], 1) < 0) {
15         fprintf(stderr, "failed to write value!\n");
16         return -1;
17     }
18
19     close(fd);
20     return 0;
21 }

```

## 5 获得GPIO输入

```

1 static int gpio_read(int pin)
2 {
3     char path[DIRECTION_MAX];
4     char value_str[3];
5     int fd;
6
7     snprintf(path, DIRECTION_MAX, "/sys/class/gpio/gpio%d/value", pin);
8     fd = open(path, O_RDONLY);
9     if (fd < 0) {
10         fprintf(stderr, "failed to open gpio value for reading!\n");
11         return -1;
12     }
13

```



```

14     if (read(fd, value_str, 3) < 0) {
15         fprintf(stderr, "failed to read value!\n");
16         return -1;
17     }
18
19     close(fd);
20     return (atoi(value_str));
21 }

```

## 6 GPIO翻转操作

【main函数】

```

1  int main(int argc, char *argv[])
2  {
3      int i = 0;
4
5      GPIOExport(POUT);
6      GPIODirection(POUT, OUT);
7
8      for (i = 0; i < 20; i++) {
9          GPIOWrite(POUT, i % 2);
10         usleep(500 * 1000);
11     }
12
13     GPIOUnexport(POUT);
14     return(0);
15 }

```

【makefile】——此处的代码tab显示可能存在问题, 请以代码仓库为主。

```

1  # 可执行文件
2  TARGET=test
3  # 依赖目标
4  SRCS=gpio-sysfs.c
5
6  # 目标文件
7  OBJS = $(SRCS:.c=.o)
8
9  # 指令编译器和选项
10 CC=gcc
11 CFLAGS=-Wall -std=gnu99
12
13 $(TARGET):$(OBJS)
14 $(CC) -o $@ $^
15
16 clean:
17 rm -rf $(TARGET) $(OBJS)
18
19 # 连续动作, 先清除再编译链接, 最后执行
20 exec:clean $(TARGET)
21 @echo 开始执行
22 sudo ./$(TARGET)
23 @echo 执行结束
24
25 # 编译规则 $@代表目标文件 $< 代表第一个依赖文件
26 %.o:%.c
27 $(CC) $(CFLAGS) -o $@ -c $<

```

### 【上传树莓派中 编译链接并执行】

make exec

makefile中exec目标包括以下一个过程, 先清除目标文件和可执行文件, 然后进行交叉编译, 最后使用 **超级权限** 运行可执行文件。

makefile的使用详见【 [Linux学习笔记——例说makefile 索引博文](#)】

## 7 总结

【1】树莓派和其他嵌入式Linux开发板存在差别和联系, 树莓派同样可以使用sysfs控制GPIO。

【2】树莓派即在其他助剂中交叉编译, 也可在平台直接编译。

## 8 参考资料

【1】 [RPI Low-level peripherals](#)



微信



支付宝

如果你觉得这篇博文对你有帮助,  
请为徐凯**打个赏**, 这将会鼓励我写出更加**优秀的博文**, 帮助更多的  
**物联网工程师、智能硬件爱好者**。

### 树莓派学习笔记——wiringPi GPIO使用详解

物联网 IoT 经验分享 8万+

最近认真学习了**树莓派**, 从浅到深认真分析了wiringPi实现代码, 借助**树莓派学习linux**收获颇丰。深入学习linux一段时间后, 发现它非常有魅力, 一个简单的I...

### sysfs方式访问gpio

qq\_41546323的博客 1263

通过**sysfs方式**控制GPIO, 先访问/sys/class/gpio目录, 向export文件写入GPIO编号, 使得该GPIO的操作接口从内核空间暴露到用户空间, GPIO的操作接...



请发表有价值的评论, 博客评论欢迎灌水, 良好的社区氛围需大家一起维护。

评论



cXspark: 你好, 我想使用按键中断控制GPIO, 请问一下如何用C实现呢? 4年前 回复 ...



jiamianshiye: 楼主你好, 有个问题想请教下。我也是在sysfs下对gpio进行读写、配置中断。但是在sysfs下, 好像没有gpio消抖, 请问在应用层对gpio消抖采用什么方法比较好? 多谢了。 6年前 回复 ...



jiamianshiye 回复 xukai871105: 多谢回复。目前我用的不是树莓派, 是TI的8168芯片。在sysfs对gpio进行捕捉中断时, 用的是poll函数。但是经常会出现 给一个按键中断, 但是却捕捉到3-5个中断。并且, 捕捉中断的程序开始运行时, 也会提示 获得了一个中断。很是头疼。 6年前 回复 ...



xukai871105 [博主] 回复: 应该没有GPIO消抖, 我觉得你可以展示不关心这个事情, 因为树莓派单纯处理GPIO的话, 比单片机要慢不少, 说不定没有重复触发的情况。 6年前 回复 ...



孤舟囊笠不倒翁: wiringPi有外部中断, 但是我不知道如何在linux上定时一秒, 从而统计一秒内发生的上升沿中断的次数(也就是方波频率) 6年前 回复 ...

登录 查看 7 条热评

### /sys/class/gpio 操作方法

viewsky11的专栏 3299

在嵌入式设备中对GPIO的操作是最基本的操作。一般的做法是写一个单独驱动程序, 网上大多数的例子都是这样的。其实linux下面有一个通用的GPIO操...

### 树莓派 ubuntu gpio\_如何给树莓派安装操作系统

weixin\_39951419的博客 42

新的**树莓派**拿到手中, 你迫不及待的要玩。别急, 这可不像你买的Macbook开机就能用, 想要顺利启动**树莓派**, 还要经历五个步骤。准备一张micro SD...

xukai871105 1 专家  
码龄14年 暂无认证

203 8402 46万+ 512万+  
原创 周排名 总排名 访问 等级

2万+ 4634 1376 1687 2534  
积分 粉丝 获赞 评论 收藏

私信

关注

搜博文文章

### 联系方式

E-mail: 172341033@qq.com

所在地: 江苏无锡

### 热门文章

MQTT学习笔记——MQTT协议体验  
Mosquitto安装和使用 225549

Linux学习笔记——如何使用echo指令向文件写入内容 190879



树莓派学习笔记——GPIO功能学习

163616

Wireshark学习笔记——如何快速抓取HTTP数据包

143927

CoAP协议学习——CoAP基础

98224

最新评论

Chrome浏览器安装CoAP调试客户端Cop...  
金桑: 博主你好, 按照您这篇文章安装好了Copper插件后点击GET, 在CoAP Mess...  
Contiki学习平台推荐  
Hugo\_yanghui: 都购买了纸质书, 正在学习中。。你贴的淘宝链接失效了, 能再贴...  
Contiki学习平台推荐  
xukai871105: 我已经写了两本和物联网有关的图书了, 我真的是写不动了  
MODBUS协议整理——写单个线圈寄存...  
Lagrange Zu: 楼主, 您写的写入线圈 ( 05H ) 的例子, 那个CRC校验码写错了吧? 应...  
Contiki学习平台推荐  
Hugo\_yanghui: 你好, 你现在知道链接了吗, 贴一下呗, 谢谢

您愿意向朋友推荐“博客详情页”吗?

🙄

😞

😐

😊

😄

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

Windows下通过环境变量设置多个KubeConfig配置文件  
Windows中安装和配置多个Python3环境  
Visual Studio Code中指定netcore启动类

2020年 3篇

2019年 5篇

2018年 1篇

2017年 3篇

2016年 6篇

2015年 9篇

2014年 103篇

2013年 70篇

2012年 8篇

2011年 1篇

树莓派使用文件IO操作GPIO SysFs方式\_huayucong的博客

10-9

树莓派使用文件IO操作GPIO SysFs方式 0 前言 本文描述如果通过文件IO sysfs方式控制树莓派 GPIO端口。通过sysfs方式控制GPIO,先访问/sys/class/gpi...

树莓派学习笔记 ——I2C PCF8574 SysFs方式\_物联网 IoT...

10-1

树莓派学习笔记 ——I2C PCF8574 SysFs方式 1.前言 树莓派的GPIO端口数量有限,可通过IO扩展芯片增加GPIO的数量,使得树莓派可以适应更多的应用。...

树莓派瞎玩~4~控制GPIO

小白的部落 4971

原文地址: 树莓派瞎玩~4~控制GPIO树莓派上提供了一组GPIO ( General Purpose Input Output ,即通用输入/输出 ) 接口, 这些接口可以用于做一些电子...

树莓派 - 通过sysfs操控GPIO

feiwatson的博客 720

点亮或熄灭LED硬件上, 一个LED灯接在pi的Pin-25. 该引脚为BCM的GPIO26\$ gpio readall +-----+-----+-----+-----+---Pi 3---+-----+-----+-----+-----

树莓派学习笔记——wiringPi简介、安装和管脚说明\_Tony...

10-14

便是使用WPI\_MODE\_PINS 模式,wiringPi的IO管脚编号和BCM IO管脚编号存在一个固定映射关系,但是wiringPi其他代码中还存在wiringPiSetupSys函数.该...

树莓派学习笔记——索引博文\_物联网 IoT 经验分享-CSDN...

8-31

【GPIO】 【树莓派学习笔记——GPIO功能学习】 【树莓派学习笔记——RPi.GPIO 流水灯】 【树莓派学习笔记——Shell脚本操作GPIO】 【树莓派学习...

树莓派Zero W GPIO控制

weixin\_30823227的博客 2110

作者: 陈拓chentuo@ms.xab.ac.cn2018.06.09/2018.07.05 0. 概述 本文介绍树莓派 Zero W的GPIO控制, 并用LED看效果。 0.1 树莓派GPIO编号方式 功...

树莓派学习笔记——Shell脚本操作GPIO

互联网 837

0.前言 树莓树莓派可以无痛入门linux, 下面通过命令和shell脚本控制树莓派GPIO。通过本文可以熟悉多个linux命令, 并熟悉shell脚本编写方法, 并对IO...

linux dts gpio sysfs example

gaoguoxin2的专栏 1215

http://blog.csdn.net/gaoguoxin2/article/details/53340350 https://community.nxp.com/thread/316241 https://community.nxp.com/message/811819 1. dts ...

Linux下通过sysfs方式控制GPIO ( /sys/class/gpio )

海月汐辰 1万+

Linux下通过sysfs方式控制GPIO ( /sys/class/gpio ) 一、首先, 查看系统中有没有"/sys/class/gpio"这个文件夹, 如果没有就在编译内核的时候勾选 Device...

Ubuntu和树莓派的远程操作练习

最新发布 eininbebop的博客 64

Linux系统 ( Ubuntu和树莓派 ) 的远程操作练习 1.虚拟机设置桥接模式 打开网络编辑器 将桥接到的网卡设置为当前使用的网卡 2.xshell远程登录 要使用Xs...

EasyARM i.mx287学习笔记——文件IO方式操作GPIO

热门推荐 物联网 IoT 经验分享 1万+

本文描述如果通过文件IO sysfs方式控制EasyARM GPIO端口。通过sysfs方式控制GPIO, 先访问/sys/class/gpio目录, 向export文件写入GPIO编号, 使得...

linux中通用GPIO接口的操作--文件IO方式

qq\_38892883的博客 1292

0 前言 本文描述如果通过文件IO sysfs方式控制EasyARM GPIO端口。通过sysfs方式控制GPIO, 先访问/sys/class/gpio目录, 向export文件写入GPIO编...

Linux下用文件IO的方式操作GPIO ( /sys/class/gpio )

weixin\_42259646的博客 322

https://blog.csdn.net/lu\_embedded/article/details/53061901

Linux下创建GPIO(/sys/class/gpio )

unbroken 1354

通过sysfs方式控制GPIO, 先访问/sys/class/gpio目录, 向export文件写入GPIO编号, 使得该GPIO的操作接口从内核空间暴露到用户空间, GPIO的操作接...

linux下对/sys/class/gpio中的gpio的控制

迪迦·奥特曼 1741

在嵌入式设备中对GPIO的操作是最基本的操作。一般的做法是写一个单独驱动程序, 网上大多数的例子都是这样的。其实linux下面有一个通用的GPIO操...

©2021 CSDN 皮肤主题: 技术工厂 设计师:CSDN官方博客 返回首页

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

xukai871105

关注

👍 1

💬 7

🌟 6

👤

专栏目录