

转载

Yfw&amp;武

2019-11-16 12:42:46

954

收藏 2

版权

分类专栏：

【linux内核分析】

文章标签：

GPIO

GPIO子系统

pinctrl驱动

gpio驱动



【linux内核分析】 专栏收录该内容

0 订阅

11 篇文章

订阅专

栏

## 一 概述

Linux内核中gpio是最简单，最常用的资源(和 interrupt ,dma,timer一样)驱动程序，应用程序都能够通过相应的接口使用gpio，gpio使用0 ~ MAX\_INT之间的整数标识，不能使用负数.gpio与硬件体系密切相关的,不过linux有一个框架处理gpio，能够使用统一的接口来操作gpio。在讲gpio核心(gpiolib.c)之前先来看看gpio是怎么使用的

## 二 内核中gpio的使用

1 测试gpio端口是否合法 int gpio\_is\_valid(int number);

2 申请某个gpio端口当然在申请之前需要显示的配置该gpio端口的pinmux

int gpio\_request(unsigned gpio, const char \*label)

3 标记gpio的使用方向包括输入还是输出

/\*成功返回零失败返回负的错误值\*/

int gpio\_direction\_input(unsigned gpio);

int gpio\_direction\_output(unsigned gpio, int value);

4 获得gpio引脚的值和设置gpio引脚的值(对于输出)

int gpio\_get\_value(unsigned gpio);

void gpio\_set\_value(unsigned gpio, int value);

5 gpio当作中断口使用

int gpio\_to\_irq(unsigned gpio);

返回的值即中断编号可以传给request\_irq()和free\_irq()

内核通过调用该函数将gpio端口转换为中断，在用户空间也有类似方法

6 导出gpio端口到用户空间

int gpio\_export(unsigned gpio, bool direction\_may\_change);

内核可以对已经被gpio\_request()申请的gpio端口的导出进行明确的管理，

参数direction\_may\_change表示用户程序是否允许修改gpio的方向，假如可以

则参数direction\_may\_change为真

/\* 撤销GPIO的导出 \*/

void gpio\_unexport();

## 三 用户空间gpio的调用

一 概述

二 内核中gpio的使用

三 用户空间gpio的调用




















(1) export/unexport文件接口：

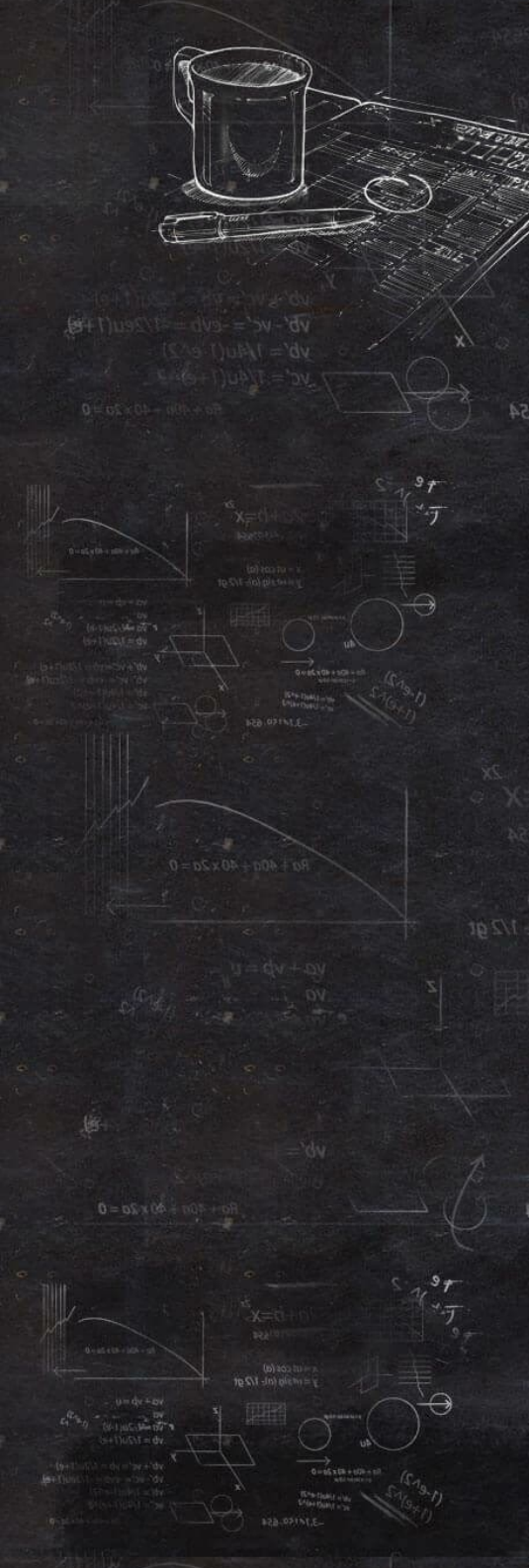
(2) /sys/class/gpio/gpioN

(3)/sys/class/gpio/gpiochipN

四 用户态使用gpio监听中断

分类专栏

	♥ Docker	5篇
	♥ wxicon开发	
	♥ PCB	
	□ 【Altium Designer】	1篇
	♥ CAD	2篇
	♥ QT	6篇
	□ 【QT移植】	2篇
	□ 【QT编程】	
	♥ uni-app	2篇
	♥ 语音识别	4篇
	♥ CAN	2篇
	♥ 蓝牙APP开发	
	♥ OpenCV	3篇
	♥ Python	16篇
	□ 【tensorflow+SSD】	4篇
	□ 【PyQt5】	7篇
	□ 【ImageAI】	3篇
	□ 【文字识别】	4篇
	□ 【人脸识别】	1篇
	♥ kormee	1篇



用户空间访问gpio，即通过sysfs接口访问gpio，下面是/sys/class/gpio目录下的三种文件：

--export/unexport文件

--gpioN指代具体的gpio引脚

--gpio\_chipN指代gpio控制器

必须知道以上接口没有标准device文件和它们的链接。

### (1) export/unexport文件接口：

/sys/class/gpio/export，该接口只能写不能读

用户程序通过写入gpio的编号来向内核申请将某个gpio的控制权导出到用户空间当然前提是没有内核代码申请这个gpio端口

比如 echo 19 > export

上述操作会为19号gpio创建一个节点gpio19，此时/sys/class/gpio目录下边生成一个gpio19的目录

/sys/class/gpio/unexport和导出的效果相反。

比如 echo 19 > unexport

上述操作将会移除gpio19这个节点。

### (2) /sys/class/gpio/gpioN

指代某个具体的gpio端口,里边有如下属性文件

direction 表示gpio端口的方向，读取结果是in或out。该文件也可以写，写入out 时该gpio设为输出同时电平默认为低。写入low或high则不仅可以

设置为输出 还可以设置输出的电平。当然如果内核不支持或者内核代码不愿意，将不会存在这个属性.比如内核调用了gpio\_export(N,0)就表示内核不愿意修改gpio端口方向属性

value 表示gpio引脚的电平,0(低电平)1（ 高电平 ）,如果gpio被配置为输出，这个值是可写的，记住任何非零的值都将输出高电平, 如果某个引脚

能并且已经被配置为中断，则可以调用poll(2)函数监听该中断，中断触发后poll(2)函数就会返回。

edge 表示中断的触发方式，edge文件有如下四个值："none", "rising", "falling", "both"。

none表示引脚为输入，不是中断引脚

rising表示引脚为中断输入，上升沿触发

falling表示引脚为中断输入，下降沿触发

both表示引脚为中断输入，边沿触发

这个文件节点只有在引脚被配置为输入引脚的时候才存在。当值是none时可以通过如下方法将变为中断引脚

echo "both" > edge;对于是both,falling还是rising依赖具体硬件的中断的触发方式。此方法即用户态gpio转换为中断引脚的方式

active\_low 不怎么明白，也未有用过

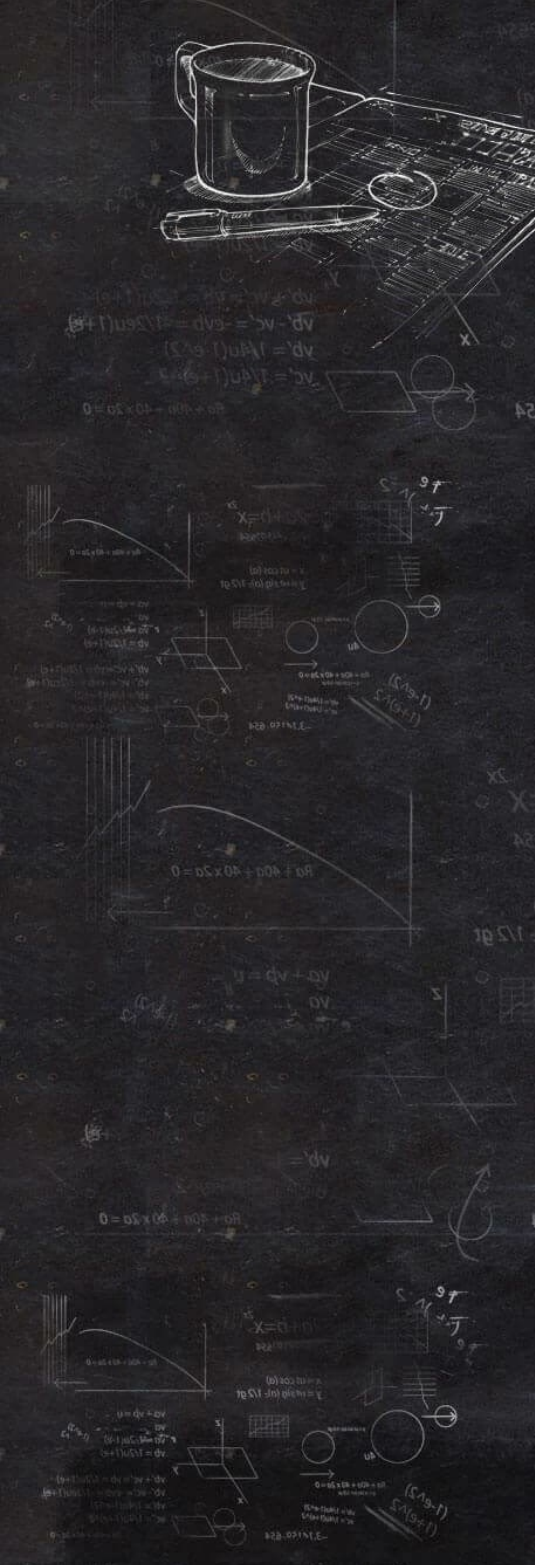
### (3)/sys/class/gpio/gpiochipN

gpiochipN表示的就是一个gpio\_chip,用来管理和控制一组gpio端口的控制器，该目录下存在一下属性文件：

base 和N相同，表示控制器管理的最小的端口编号。

	♥ PIC单片机	8篇
	□ 【PIC->C编程】	7篇
	♥ 汽车电子	1篇
	♥ CSS	1篇
	♥ CSDN	
	♥ Linux	16篇
	□ 【linux内核分析】	11篇
	□ 【LINUX应用开发】	1篇
	□ 【linux音频系统】	3篇
	□ 【Linux设备树详解】	1篇
	□ 【buildroot应用】	7篇
	□ 【LINUX驱动编程】	3篇
	□ 【wireless_tool】	6篇
	□ 【wpa_supplicant】	3篇
	□ 【kconfig】	1篇
	□ 【linux开发遇到的问...	1篇
	□ 【热插拔】	1篇
	♥ u-boot	
	□ 【u-boot实践】	3篇
	□ 【u-boot 分析】	10篇
	♥ ARM	1篇
	□ 【ARM嵌入式开发】	2篇
	□ 【ARM汇编】	1篇
	□ 【裸奔吧ARM】	1篇
	□ 【F1C100S学习】	25篇
	□ 【V3S】	19篇
	□ 【MINI2440开发板】	41篇
	♥ VB	2篇





lable 诊断使用的标志（并不总是唯一的）

ngpio 表示控制器管理的gpio端口数量（端口范围是：N ~ N+ngpio-1）

## 四 用户态使用gpio监听中断

首先需要将该gpio配置为中断

```
echo "rising" > /sys/class/gpio/gpio12/edge
```

以下是伪代码

```
int gpio_id;
```

```
struct pollfd fds[1];
```

```
gpio_fd = open("/sys/class/gpio/gpio12/value",O_RDONLY);
```

```
if( gpio_fd == -1 )
```

```
err_print("gpio open");
```

```
fds[0].fd = gpio_fd;
```

```
fds[0].events = POLLPRI;
```

```
ret = read(gpio_fd,buff,10);
```

```
if( ret == -1 )
```

```
err_print("read");
```

```
while(1){undefined
```

```
ret = poll(fds,1,-1);
```

```
if( ret == -1 )
```

```
err_print("poll");
```

```
if( fds[0].revents & POLLPRI){undefined
```

```
ret = lseek(gpio_fd,0,SEEK_SET);
```

```
if( ret == -1 )
```

```
err_print("lseek");
```

```
ret = read(gpio_fd,buff,10);
```

```
if( ret == -1 )
```

```
err_print("read");
```

```
/*此时表示已经监听到中断触发了，该干事了*/
```

```
.....
```

```
}
```

```
}
```

记住使用poll()函数，设置事件监听类型为POLLPRI和POLLERR在poll()返回后，使用lseek()移动到文件开头读取新的值或者关闭它再重新

	unix高级环境编程	1篇
	在Ubuntu上学驱动	2篇
	各linux操作系统的安装...	3篇
	arm-linux-gcc	1篇
	ADS1.2	1篇
	linux驱动开发	1篇
	tslib移植	2篇
	视频接口设计	1篇
	♥ 树莓派（raspberry...	7篇
	♥ uni-app	8篇
	Keil使用	
	♥ MySQL	10篇
	♥ Labview	4篇
	VB	7篇
	C/C++	8篇
	♥ JAVA	1篇
	PIC单片机	
	Web	3篇
	WebSocket	1篇
	♥ PHP	7篇
	♥ 二维码	1篇
	计算机原理	1篇
	FRP	2篇
	WordPress	3篇
	C#	1篇
	VS IDE的使用	1篇
	GT2440开发板	8篇
	♥ Vue.js	12篇

打开读取新值。必须这样做否则poll函数会总是返回。

通过/sys/文件接口操作IO端口

gpio\_operation 通过/sys/文件接口操作IO端口 GPIO到文件系统的映射

\* 控制GPIO的目录位于/sys/class/gpio

\* /sys/class/gpio/export文件用于通知系统需要导出控制的GPIO引脚编号

\* /sys/class/gpio/unexport 用于通知系统取消导出

\* /sys/class/gpio/gpiochipX目录保存系统中GPIO寄存器的信息，包括每个寄存器控制引脚的起始编号base，寄存器名称，引脚总数 导出一个引脚的操作步骤

\* 首先计算此引脚编号，引脚编号 = 控制引脚的寄存器基数 + 控制引脚寄存器位数

\* 向/sys/class/gpio/export写入此编号，比如12号引脚，在shell中可以通过以下命令实现，命令成功后生成/sys/class/gpio/gpio12目录，如果没有出现相应的目录，说明此引脚不可导出：

echo 12 > /sys/class/gpio/export

\* direction文件，定义输入输入方向，可以通过下面命令定义为输出

echo out > direction

\* direction接受的参数：in, out, high, low. high/low同时设置方向为输出，并将value设置为相应的1/0。

\* value文件是端口的数值，为1或0。

echo 1 > value

## 如何驱动GPIO及对应设备

11-13

一般的ARM都提供超过100路的GPIO复用管脚，要使用这些管脚，就必须首先将其驱动起来。如果使用的芯片带有MMU内存管理，那么在写驱动模块的时...

### Linux驱动开发（十一）：pinctrl系统和GPIO子系统

a568713197的博客 1230

pinctrl系统和GPIO子系统简介 简介 配置寄存器来控制IO的方式太过于原始，Linux内核提供了pinctrl系统和gpio子系统用于GPIO驱动，当然pinctrl子...



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。

抢沙发



评论

### Linux内核驱动之gpio子系统的使用

华清远见 | 程序员的学习天堂 736

linux内核中gpio是最简单，最常用的资源(和 interrupt ,dma,timer一样)驱动程序，应用程序都能够通过相应的接口使用gpio，gpio使用0 ~ MAX\_INT之间的...

### 【Linux基础系列之】gpio系统

ForDream321的专栏 1049

上一章介绍了pinctrl系统管理所有的物理pin脚，gpio也是pin脚的一种，所以需要某个gpio的时候就需要通过pinctrl把某个pin脚设置为gpio功能就即可；在...

### Linux内核驱动之GPIO子系统-GPIO的使用\_one shot,one k...

10-27

Linux内核驱动之GPIO子系统-GPIO的使用 Linux内核中gpio是最简单,最常用的资源(和interrupt ,dma,timer一样)驱动程序,应用程序都能够通过相应的接口...



Yfw&武  
码龄8年

暂无认证

239

14万+

9040

58万+



原创

周排名

总排名

访问

等级

7994

135

207

115

1193

积分

粉丝

获赞

评论

收藏



私信

关注

搜博文章



热门文章



Visual Studio Code 使...

3篇



♥ Electron

8篇



Git的安装

1篇



Git

1篇



♥ Element

2篇



学习方法

1篇



♥ Javascript

5篇



JavaWeb

7篇



数据结构与算法

1篇



数据库

1篇



Servlet教程

5篇



Tomcat

1篇



JSP

8篇



SpringMVC

5篇



log4j

1篇



Node.js

2篇



redis



什么是EPEL?及epel release的安装

25217

修改el-input样式

19585

修改el-button的样式

16131

uni-app 整包升级/更新方案

14736

python3安装后没有pip解决

14669

最新评论

JSP开发的工具下载与环境搭建 ( Windo...  
m0\_59359928: 想知道为什么我的tomcat绑定后黄色小字只有stopped没有republics...

python写按键 ( Iradc ) 录音程序  
年年618: 查了一下 from evdev import Input Device仅限linux , Windows怎么操作呢

Linux下安装VSCode  
NPU\_QY: 第六步如果vim中 E212:无法打开并写入文件的话 , 把vim前面加个sudo

荔枝派 zero(全志V3S)-编译及SD烧录  
QQ\_526642524: 正在学习中。 . . . . .  
. . .

u-boot 添加LED驱动 ( for F1C100S )  
洛拓: 博主 , 为啥我想您这样配置之后电平无变化呢 , crt0.S文件是位于什么目录下...

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

Vue如何支持JSX语法

关于在electron-vue项目中使用ref

VB向DLL传递字符串数组的问题

2021年 15篇

2020年 70篇

2019年 288篇

linux驱动开发学习笔记十五:gpio子系统介绍\_耐心的小黑...

10-26

主要目的就是方便驱动开发者使用 gpio,驱动开发者在设备树中添加 gpio 相关信息,然后就可以在驱动程序中使用 gpio 子系统提供的 API函数来操作 GPIO,...

linux中通用GPIO接口的操作

森林中的波罗 3365

转载地址：http://blog.csdn.net/cjyusha/article/details/50418862 在linux嵌入式设备开发中，对GPIO的操作是最常用的，在一般的情况下，一般都有对...

linux内核gpio编号,Linux内核驱动之GPIO子系统(一)GPIO的使用

weixin\_31434215的博客 131

一概述Linux内核中gpio是最简单，最经常使用的资源(和interrupt,dma,timer同样)驱动程序，应用程序都可以经过相应的接口使用gpio，gpio使用0 ~ MAX\_...

linux设备驱动 ( GPIO子系统 )

rookie的专栏 4701

一、gpio子系统的内核源码主要函数 源代码：Gpiolib.c (arch/arm/mach-s5pv210) static \_\_init int s5pv210\_gpiolib\_init(void) { struct s3c\_gpio\_chip \*chi...

Linux下/sys/class/gpio驱动调试

qq\_29239993的博客 751

1、gpio类的export、unexport/属性来源和作用 1.1export和unexport/属性创建 在内核初始化阶段，将自动调用gpiolib\_sysfs\_init()(在/sys/class中创建gpio...

linux中通用GPIO接口的操作--文件IO方式

qq\_38892883的博客 1293

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 下的gpio驱动程序,LinuxGPIO驱动使用其实很简单

最新发布 weixin\_39526459的博客 43

Contents[hide]ReferencesGPIO Usage from a Linux ApplicationOverviewThe following table summarizes the steps to configuring and using GPIO signa...

Linux GPIO用户空间操作

风之伤 6590

目前遇到一个问题，内核中已实现了gpio驱动，而且在内核空间可以正常使用 gpio\_request,gpio\_direction\_output,gpio\_free等gpiolib中提供的函数 ...

Linux 内核设备驱动之GPIO驱动之GPIO GPIO描述符到GPIO号

星空探索 959

/\* \* Convert a GPIO descriptor to the integer namespace. \* This should disappear in the future but is needed since we still \* use GPIO numbers for e...

LinuxGPIO驱动使用其实很简单

DriverMonkey的专栏 1万+

Author:DriverMonkey Mail:bookworepeng@Hotmail.com Phone:13410905075 QQ:196568501 本文资料主要来自互联网，并通过本人亲测试正确性 How...

linux通用GPIO驱动

zjy900507的博客 1458

Linux开发平台实现了通用GPIO的驱动，用户通过，Shell或者系统调用能控制GPIO的输出和读取其输入值。其属性文件均在/sys/class/gpio/目录下，该目...

Linux GPIO 驱动 ( gpiolib )

StephenZhou 1万+

目录 1、简述 2、Gpiolib 相关数据结构分析 2.1 gpio\_chip 结构 2.2 gpio\_desc 结构 2.3 gpio\_device 结构 3、Gpiolib 对接芯片底层 3.1、注册 GPIO 资源...

linux驱动开发 ( 一 )—GPIO驱动框架

热门推荐 yang blog 2万+

前言 GPIO驱动是Linux驱动开发中最基础、但却是很常用、很重要的驱动。比如你要点亮一个LED灯、键盘扫描、输出高低电平等等。而Linux内核...

linux驱动 ( 第十八课, PINCTRL, GPIO, GPIO\_CHIP, IRQ\_CHIP )

Huskar\_Liu的博客 503

在linux中,pinctrl被定义为一个子系统。pinmux功能被pinctrl子系统所管理。这个功能通常有SOC厂商提供。我们需要关心的是，如何使用它的功能。pin\_...

Linux GPIO 的注册与申请

江风的专栏 1万+

Linux GPIO 的注册与申请 Linux Kernel, GPIO, ARM 在Linux kernel代码中，经常会使用 GPIO 来作为一个特殊的信号，如作为芯片的片选信号等。 GPIO ...

©2021 CSDN 皮肤主题: 技术黑板 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

Yfw&武

关注

👍 0

👎 0

💬 0

🌟 2

👤

🔖

专栏目录