



[www.enjoylinux.cn](http://www.enjoylinux.cn)

# LINUX

## 硬件设备访问



版权声明：本课件及其印刷物、视频的版权归成都国嵌信息技术有限公司所有，并保留所有权力：任何单位或个人未经成都国嵌信息技术有限公司书面授权，不得使用该课件及其印刷物、视频从事商业、教学活动。已经取得书面授权的，应在授权范围内使用，并注明“来源：国嵌”。违反上述声明者，我们将追究其法律责任。

# Contents



mmap设备操作

硬件访问

混杂设备驱动

LED驱动程序设计

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# Contents



mmap设备操作

硬件访问

混杂设备驱动

LED驱动程序设计

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# mmap系统调用 (功能)



[www.enjoylinux.cn](http://www.enjoylinux.cn)

```
void* mmap ( void * addr , size_t len , int prot , int flags ,  
int fd , off_t offset )
```

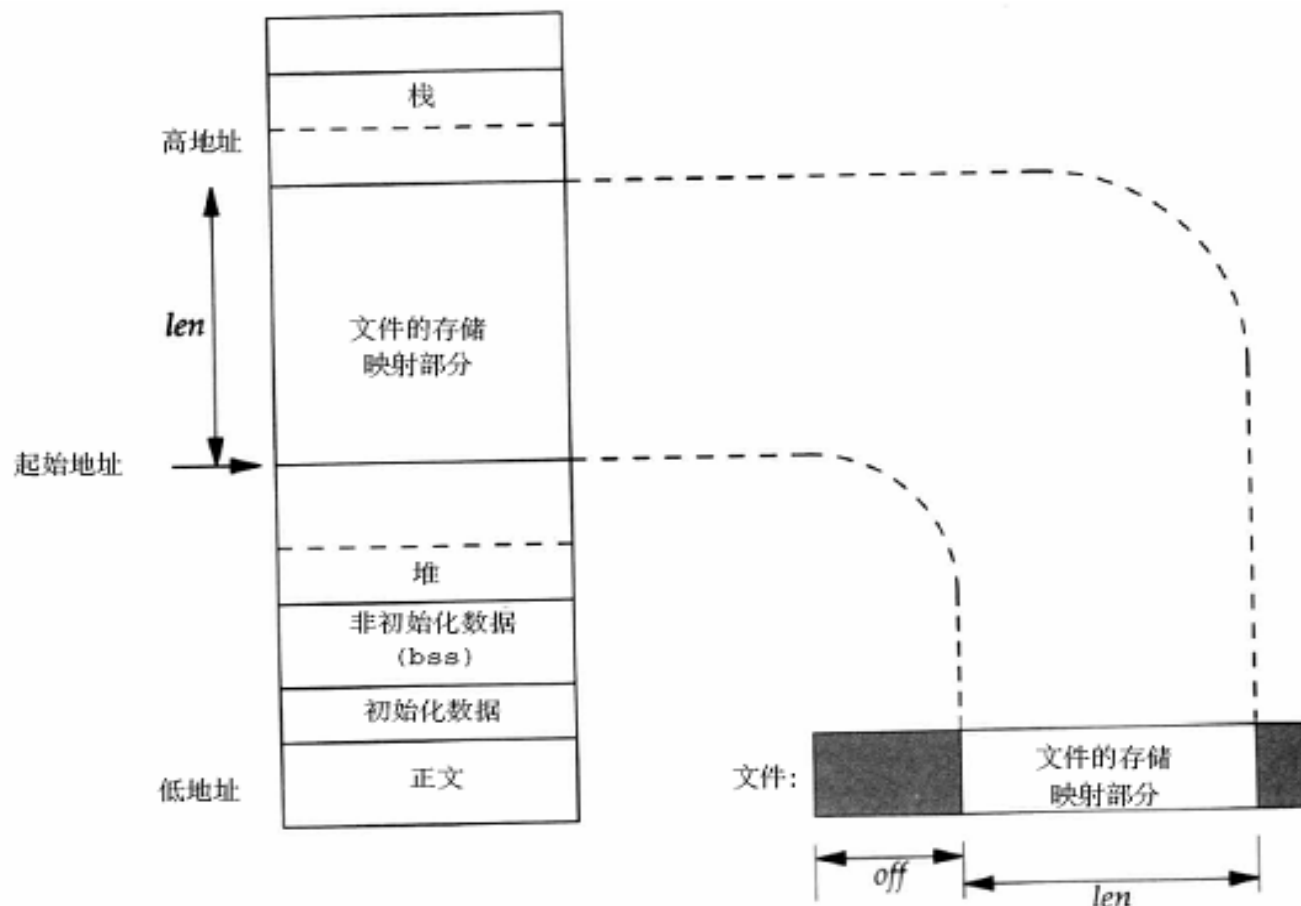
内存映射函数mmap, 负责把**文件内容映射到进程的虚拟内存空间**, 通过对这段内存的读取和修改, 来实现对文件的读取和修改, 而不需要再调用read, write等操作。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116





# mmap系统调用 (功能)



嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# mmap系统调用 (参数)



## ✓ **addr:**

指定映射的起始地址, 通常设为NULL, 由系统指定。

## ✓ **length:**

映射到内存的文件长度。

## ✓ **prot:**

映射区的保护方式, 可以是:

**PROT\_EXEC:** 映射区可被执行

**PROT\_READ:** 映射区可被读取

**PROT\_WRITE:** 映射区可被写入

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# mmap系统调用 (参数)



**flags:** 映射区的特性, 可以是:

✓ **MAP\_SHARED:**

写入映射区的数据会复制回文件, 且允许其他映射该文件的进程共享。

✓ **MAP\_PRIVATE:**

对映射区的写入操作会产生一个映射区的复制 (copy-on-write), 对此区域所做的修改不会写回原文件。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# mmap系统调用 (参数)



**fd:**

由open返回的文件描述符, 代表要映射的文件。

**offset:**

以文件开始处的偏移量, 必须是分页大小的整数倍, 通常为0, 表示从文件头开始映射。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116





# 解除映射



```
int munmap(void *start, size_t length)
```

## 功能:

取消参数start所指向的映射内存，参数length表示欲取消的内存大小。

## 返回值:

解除成功返回0，否则返回-1，错误原因存于errno中。

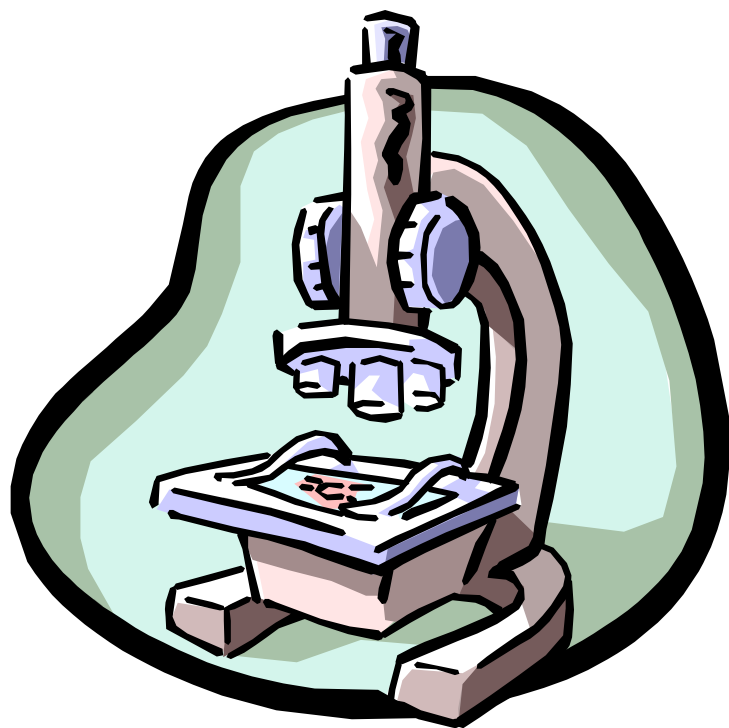
嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 实例分析



[www.enjoylinux.cn](http://www.enjoylinux.cn)



## mmap系统调用

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# 虚拟内存区域



虚拟内存区域是进程的**虚拟地址空间**中的一个**同质区间**，即具有**同样特性的连续地址范围**。一个进程的内存映象由下面几部分组成：**程序代码、数据、BSS和栈区域**，以及**内存映射的区域**。

嵌入式Linux技术咨询QQ号：550491596  
嵌入式Linux学习交流QQ群：65212116



# 虚拟内存区域



[www.enjoylinux.cn](http://www.enjoylinux.cn)

一个进程的内存区域可以通过查看  
**/proc/pid/maps**

```
08048000-0804f000 r-xp 00000000 08:01 573748 /sbin/rpc.statd #text
0804f000-08050000 rw-p 00007000 08:01 573748 /sbin/rpc.statd #data
08050000-08055000 rwxp 00000000 00:00 0 #bss
040000000-40015000 r-xp 00000000 08:01 933965 /lib/ld2.3.2.so #text
40015000-40016000 rw-p 00014000 08:01 933965 /lib/ld-2.3.2.so #data
```

**嵌入式Linux技术咨询QQ号: 550491596**  
**嵌入式Linux学习交流QQ群: 65212116**



# 虚拟内存区域



[www.enjoylinux.cn](http://www.enjoylinux.cn)

每一行的域为:

**start\_end perm offset major:minor inode**

- ✓ **Start:** 该区域起始虚拟地址
- ✓ **End:** 该区域结束虚拟地址
- ✓ **Perm:** 读、写和执行权限; 表示对这个区域, 允许进程做什么。这个域的最后字符要么是p表示私有的, 要么是s表示共享的。
- ✓ **Offset:** 被映射部分在文件中的起始地址
- ✓ **Major、minor:** 主次设备号
- ✓ **Inode:** 索引结点

**嵌入式Linux技术咨询QQ号: 550491596**

**嵌入式Linux学习交流QQ群: 65212116**





# vm\_area\_struct



[www.enjoylinux.cn](http://www.enjoylinux.cn)

Linux内核使用结构**vm\_area\_struct**

(`<linux/mm_types.h>`) 来描述虚拟内存区域，其中几个主要成员如下：

✓ **unsigned long vm\_start**

虚拟内存区域起始地址

✓ **unsigned long vm\_end**

虚拟内存区域结束地址

嵌入式Linux技术咨询QQ号：550491596

嵌入式Linux学习交流QQ群：65212116



# vm\_area\_struct



[www.enjoylinux.cn](http://www.enjoylinux.cn)

## ✓ unsigned long vm\_flags

该区域的标记。如:VM\_IO和VM\_RESERVED。

VM\_IO将该VMA标记为内存映射的IO区域，  
VM\_IO会阻止系统将该区域包含在进程的存放转存(core dump)中，VM\_RESERVED标志内存区域不能被换出。

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# mmap设备操作



映射一个设备是指把用户空间的一段地址关联到设备内存上。当程序读写这段用户空间的地址时，它实际上是在访问设备。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# mmap设备操作



## mmap设备方法需要完成什么功能？

mmap方法是file\_operations结构的成员，在mmap系统调用发出时被调用。在此之前，内核已经完成了很多工作。mmap设备方法所需要做的就是**建立虚拟地址到物理地址的页表**。

```
int (*mmap) (struct file *, struct vm_area_struct *)
```

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# mmap设备操作



mmap如何完成页表的建立?

方法有二:

- 1、使用 **remap\_pfn\_range** 一次建立所有页表;
- 2、使用 **nopage VMA** 方法每次建立一个页表。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116





# mmap设备操作



构造页表的工作可由remap\_pfn\_range函数完成，原型如下：

```
int remap_pfn_range(struct
    vm_area_struct *vma, unsigned long
    addr, unsigned long pfn, unsigned long
    size, pgprot_t prot)
```

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# remap\_page\_range



[www.enjoylinux.cn](http://www.enjoylinux.cn)

**vma:**

虚拟内存区域指针

**virt\_addr:**

虚拟地址的起始值

**pfn:**

要映射的物理地址所在的物理页帧号，可将物理地址  
>>PAGE\_SHIFT得到。

**size:**

要映射的区域的大小。

**prot:**

VMA的保护属性。

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# mmap设备操作



[www.enjoylinux.cn](http://www.enjoylinux.cn)

```
int memdev_mmap(struct file*filp, struct vm_area_struct *vma)
{
    Vma->vm_flags |= VM_IO;
    Vma->vm_flags |= VM_RESERVED;

    if (remap_pfn_range(vma, vma->vm_start,
        virt_to_phys(dev->data)>> PAGE_SHIFT,
        size,
        vma->vm_page_prot))
        return -EAGAIN;
    return 0;
}
```

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 实验



[www.enjoylinux.cn](http://www.enjoylinux.cn)

## mmap设备方法实现

§ 实现字符驱动程序的mmap设备方法

§ 实现测试应用程序

(在mini2440平台完成上述实验)

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# Contents



mmap设备操作

硬件访问

混杂设备驱动

LED驱动程序设计

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116





# 寄存器与内存



寄存器与内存的区别在哪里呢？

寄存器和 RAM 的主要不同在于寄存器操作有副作用（**side effect** 或 边际效果）：  
读取某个地址时可能导致该地址内容发生变化，比如很多设备的中断状态寄存器只要一读取，便自动清零。

嵌入式Linux技术咨询QQ号：550491596  
嵌入式Linux学习交流QQ群：65212116



# 内存与I/O



[www.enjoylinux.cn](http://www.enjoylinux.cn)

在X86处理器中存在**I/O空间**的概念，I/O空间是相对**内存空间**而言的，他们是彼此独立的地址空间，在32位的x86系统中，I/O空间大小为**64K**，内存空间大小为**4G**。

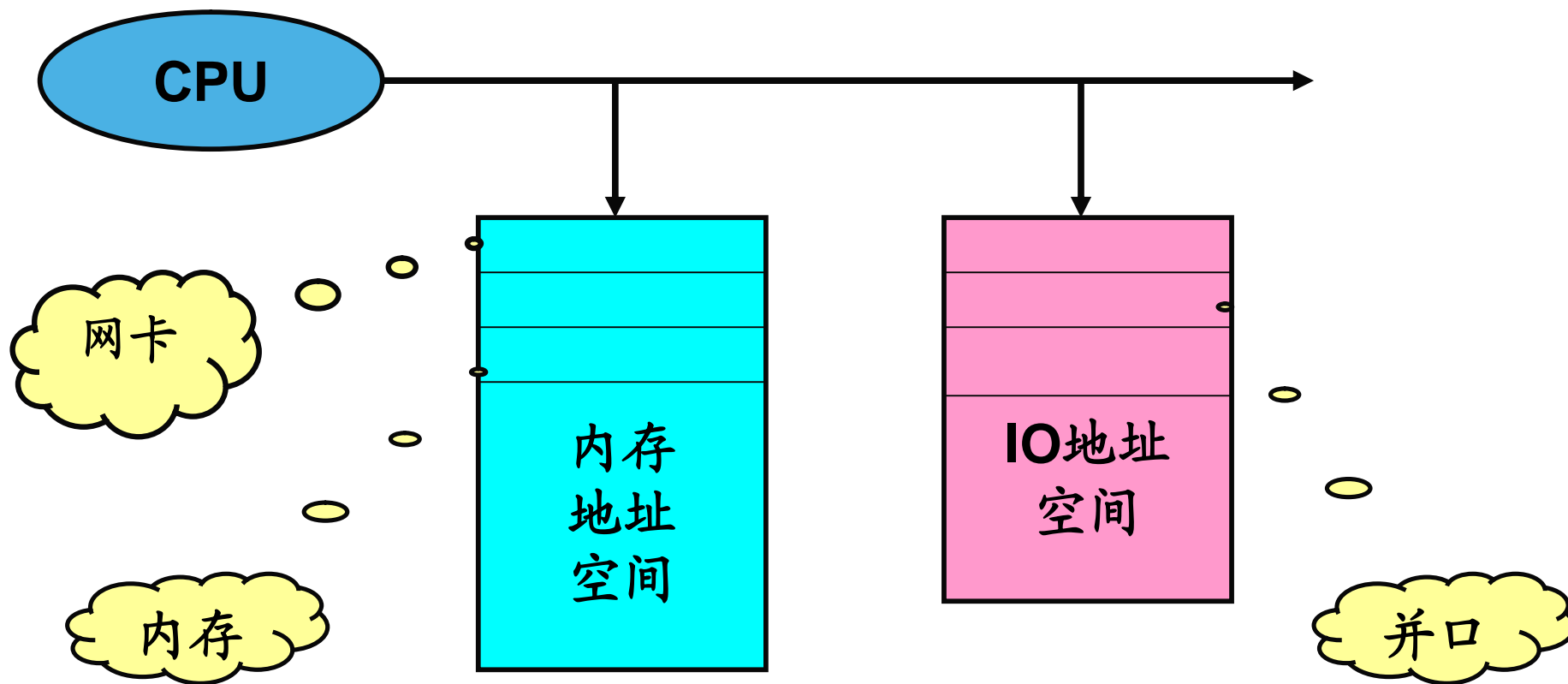
嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 内存与I/O



[www.enjoylinux.cn](http://www.enjoylinux.cn)



嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# IO/内存空间



## ✓ X86

支持内存空间、IO空间

## ✓ ARM

只支持内存空间

## ✓ MIPS

只支持内存空间

## ✓ PowerPC

只支持内存空间

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# IO端口与IO内存



[www.enjoylinux.cn](http://www.enjoylinux.cn)

## √IO端口:

当一个寄存器或内存位于**IO空间**时，称其为**IO端口**。

## √IO内存:

当一个寄存器或内存位于**内存空间**时，称其为**IO内存**。

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116





# 操作I/O端口



[www.enjoylinux.cn](http://www.enjoylinux.cn)

对I/O端口的操作需按如下步骤完成:

1. 申请

2. 访问

3. 释放

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# 申请I/O端口



[www.enjoylinux.cn](http://www.enjoylinux.cn)

内核提供了一套函数来允许驱动申请它需要的I/O端口，其中核心的函数是：

```
struct resource *request_region(unsigned long first,  
                                unsigned long n, const char *name)
```

这个函数告诉内核，你要使用从 **first** 开始的 **n** 个端口，**name** 参数是设备的名字。如果申请**成功**，返回**非 NULL**，申请**失败**，返回 **NULL**。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 申请I/O端口



[www.enjoylinux.cn](http://www.enjoylinux.cn)

系统中端口的分配情况记录在  
**/proc/ioproports** 中(展示)。如果不能分配  
需要的端口，可以来这里查看谁在使  
用。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 访问I/O端口



I/O端口可分为8-位, 16-位, 和 32-位端口。Linux内核头文件(体系依赖的头文件 `<asm/io.h>`) 定义了下列内联函数来访问 I/O 端口:

- **unsigned inb(unsigned port)**  
读字节端口( 8 位宽 )
- **void outb(unsigned char byte, unsigned port)**  
写字节端口( 8 位宽 )。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 访问I/O端口



[www.enjoylinux.cn](http://www.enjoylinux.cn)

✓ unsigned **inw**(unsigned port)

✓ void **outw**(unsigned short word, unsigned port)

存取 16-位 端口。

✓ unsigned **inl**(unsigned port)

✓ void **outl**(unsigned longword, unsigned port)

存取 32-位 端口。

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# 释放I/O端口



[www.enjoylinux.cn](http://www.enjoylinux.cn)

当用完一组 I/O 端口(通常在驱动卸载时), 应使用如下函数把它们返还给系统:

`void release_region(unsigned long start, unsigned long n)`



嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116





# 操作I/O内存



对I/O内存的操作需按如下步骤完成:

1. 申请
2. 映射
3. 访问
4. 释放

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 申请I/O内存



[www.enjoylinux.cn](http://www.enjoylinux.cn)

内核提供了一套函数来允许驱动申请它需要的I/O内存，其中核心的函数是：

```
struct resource *request_mem_region(unsigned  
long start, unsigned long len, char *name)
```

这个函数申请一个从start 开始,长度为len 字节的内存区。如果成功,返回非NULL；否则返回NULL，所有已经在使用的I/O内存存在 /proc/iomem 中列出。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 映射I/O内存



[www.enjoylinux.cn](http://www.enjoylinux.cn)

在访问I/O内存之前, 必须进行物理地址到虚拟地址的映射, **ioremap** 函数具有此功能:

```
void *ioremap(unsigned long phys_addr, unsigned long size)
```



嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 访问I/O内存



[www.enjoylinux.cn](http://www.enjoylinux.cn)

访问 I/O 内存的正确方法是通过一系列内核提供的函数:

✓ 从 I/O 内存读, 使用下列之一:

unsigned **ioread8**(void \*addr)

unsigned **ioread16**(void \*addr)

unsigned **ioread32**(void \*addr)

✓ 写 I/O 内存, 使用下列之一:

void **iowrite8**(u8 value, void \*addr)

void **iowrite16**(u16 value, void \*addr)

void **iowrite32**(u32 value, void \*addr)

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# 访问I/O内存



[www.enjoylinux.cn](http://www.enjoylinux.cn)

老版本的 I/O 内存访问函数:

✓ 从 I/O 内存读, 使用下列之一:

unsigned **readb**(address)

unsigned **readw**(address)

unsigned **readl**(address)

✓ 写 I/O 内存, 使用下列之一:

unsigned **writeb**(unsigned value, address)

unsigned **writew**(unsigned value, address)

unsigned **writel**(unsigned value, address)

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



# 释放I/O内存



[www.enjoylinux.cn](http://www.enjoylinux.cn)

I/O内存不再需要使用时应当释放,步骤如下:

1. void **iounmap**(void \* addr)
2. void **release\_mem\_region**(unsigned long start, unsigned long len)

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116





# Contents



mmap设备操作

硬件访问

混杂设备驱动

LED驱动程序设计

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 定义



[www.enjoylinux.cn](http://www.enjoylinux.cn)

在Linux系统中，存在一类**字符设备**，它们**共享一个主设备号（10）**，但次设备号不同，我们称这类设备为**混杂设备 (miscdevice)**。所有的混杂设备形成一个链表，对设备访问时内核根据**次设备号**查找到相应的**miscdevice**设备。

嵌入式Linux技术咨询QQ号：550491596  
嵌入式Linux学习交流QQ群：65212116



# 设备描述



Linux内核使用**struct miscdevice**来描述  
一个混杂设备。

```
struct miscdevice {  
    int minor; /* 次设备号*/  
    const char *name; /* 设备名*/  
    const struct file_operations *fops; /*文件操作*/  
    struct list_head list;  
    struct device *parent;  
    struct device *this_device;  
};
```

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 设备注册



Linux内核使用**misc\_register**函数来注册一个混杂设备驱动。

```
int misc_register(struct miscdevice * misc)
```



嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# Contents



mmap设备操作

硬件访问

混杂设备驱动

LED驱动程序设计

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 上拉/下拉电阻



[www.enjoylinux.cn](http://www.enjoylinux.cn)

**上拉**是将不确定的信号通过一个电阻**与电源相连，固定在高电平**。**下拉**是将不确定的信号通过一个电阻**与地相连，固定在低电平**。上拉是对器件注入电流，下拉是输出电流。当一个接有上拉电阻的I/O端口设为输入状态时，它的常态为高电平，可用于检测低电平的输入。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116





# S3c2440 I/O端口



S3C2440包含GPA、GPB、.....、GPJ 九组 I/O端口。它们的寄存器是相似的：**GPxCON**用于设置端口功能(00 表示输入、01表示输出、10表示特殊功能、11 保留不用)，**GPxDAT** 用于读/写数据，**GPxUP** 用于决定是否使用内部上拉电阻(某位为0 时，相应引脚无内部上拉；为1时，相应引脚使用内部上拉)。

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 引脚



[www.enjoylinux.cn](http://www.enjoylinux.cn)

## LED与IO端口引脚的对应关系

**1.LED1 GPB5**

**2.LED2 GPB6**

**3.LED3 GPB7**

**4.LED4 GPB8**

**嵌入式Linux技术咨询QQ号: 550491596**

**嵌入式Linux学习交流QQ群: 65212116**



# 实例分析



## LED驱动程序



嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116



# 实验



[www.enjoylinux.cn](http://www.enjoylinux.cn)

## LED驱动程序设计

- n 实现命令控制灯全亮、全灭  
(在mini2440平台完成上述实验)

嵌入式Linux技术咨询QQ号: 550491596  
嵌入式Linux学习交流QQ群: 65212116

