



www.enjoylinux.cn

LINUX

网卡触摸屏驱动程序

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



Linux网络体系架构

网卡驱动设计

Dm9000网卡驱动分析

Linux输入子系统

触摸屏驱动设计

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



Contents



www.enjoylinux.cn



Linux网络体系架构

网卡驱动设计

Dm9000网卡驱动分析

Linux输入子系统

触摸屏驱动设计

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



体系结构



需要掌握的知识点:

1. Linux的协议栈层次
2. Linux的网络子系统架构



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



协议栈



Linux的优点之一在于它**丰富而稳定**的网络协议栈。其范围从协议无关层（例如通用 **socket** 层接口或设备层）到各种具体的网络协议实现。

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



协议简介



对于网络的理论介绍一般都采用 OSI（Open Systems Interconnection）模型，但是 **Linux** 中网络栈的介绍一般分为**四层的 Internet 模型**。

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



协议栈层次对比



OSI七层网络模型	Linux TCP/IP 四层概念模型	对应网络协议
应用层	应用层	TFTP, FTP, NFS, WAIS
表示层		Telnet, Rlogin, SNMP, Gopher
会话层		SMTP, DNS
传输层	传输层	TCP, UDP
网络层	网际层	IP, ICMP, ARP, RARP, AKP, UUCP
数据链路层	网络接口	FDDI, Ethernet, Arpanet, PDN, SLIP, PPP
物理层		IEEE 802.1A, IEEE 802.2

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



网络接口层



网络接口层把数据链路层和物理层合并在一起，提供访问物理设备的驱动程序，对应的网络协议主要是以太网协议。

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



网际层



www.enjoylinux.cn

网络层协议**管理**离散的计算机间的数据**传输**，如**IP**协议为用户和远程计算机提供了信息包的传输方法，确保信息包能正确地到达目的机器。重要的网络层**协议**包括**ARP**（地址解析协议）、**ICMP**（Internet控制消息协议）和**IP**协议（网际协议）等。

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



传输层



传输层的功能包括：格式化信息流、提供可靠传输。传输层包括**TCP**（Transmission Control Protocol，传输控制协议）和**UDP**（User Datagram Protocol，用户数据报协议），它们是传输层中最主要的协议。

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



应用层



应用层位于**协议栈的顶端**，它的主要任务是**服务于应用**，如利用**FTP**（文件传输协议）传输一个文件。常见的应用层协议有：**HTTP，FTP，Telnet**等。应用层是Linux网络设定很关键的一层，Linux服务器的配置文档主要针对应用层中的协议。

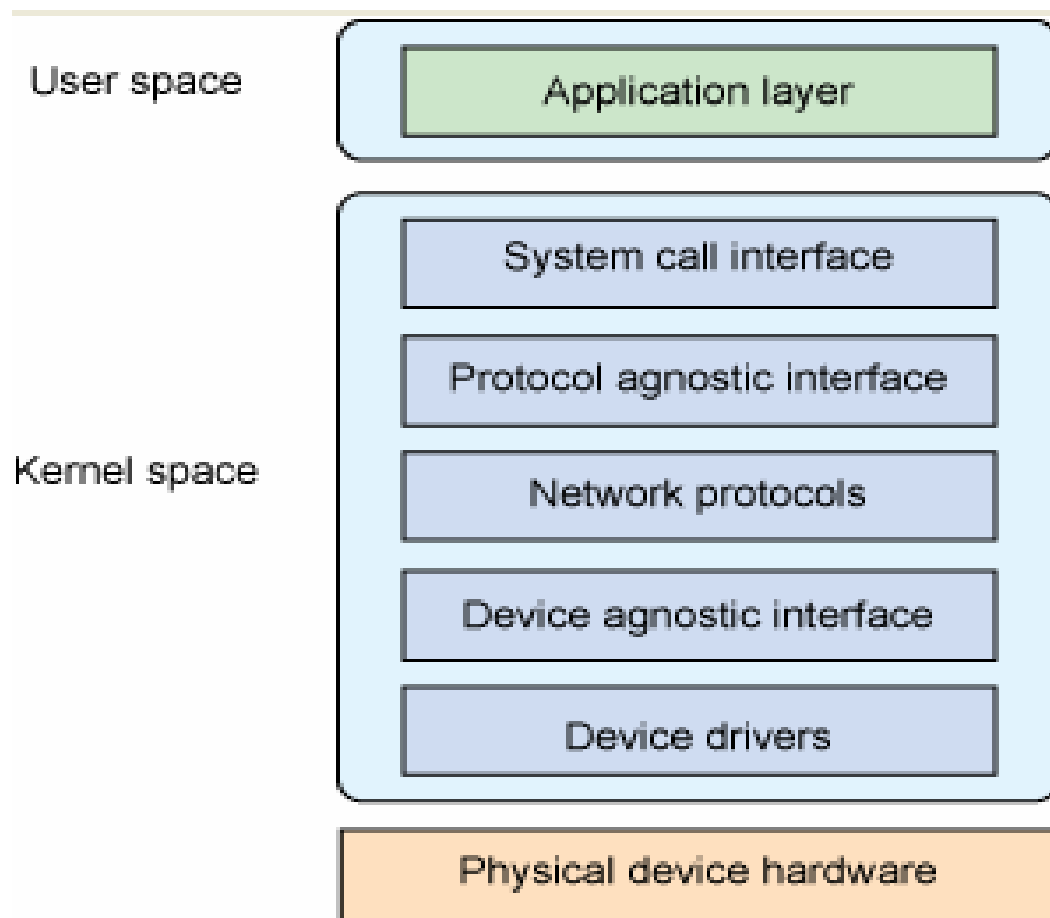
嵌入式Linux技术咨询QQ号：550491596
嵌入式Linux学习交流QQ群：65212116



Linux网络子系统



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



Linux网络子系统



www.enjoylinux.cn

Linux 网络子系统的顶部是**系统调用接口层**。它为用户空间的应用程序提供了一种访问内核网络子系统的方法。位于其下面的是一个**协议无关层**，它提供了一种通用方法来使用传输层协议。然后是**具体协议的实现**，在 Linux 中包括内嵌的协议 TCP、UDP，当然还有 IP。然后是**设备无关层**，它提供了协议与设备驱动通信的通用接口，最下面是**设备驱动程序**。

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



系统调用接口



为应用程序提供访问内核网络子系统的方法：**Socket**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



协议无关接口



实现一组**通用函数**来访问各种不同的协议：
通过socket实现。Linux 中的 **socket** 使用 **struct sock** 来描述，这个结构包含了特定 **socket** 所需要的所有状态信息，还包括 **socket** 所使用的特定协议和在 **socket** 上可以执行的一些操作。

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



网络协议



www.enjoylinux.cn

网络协议层用于实现各种具体的网络协议，如：**TCP、UDP** 等。



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



设备无关接口



设备无关接口将协议与各种网络设备驱动连接在一起。这一层提供一组通用函数供底层网络设备驱动程序使用，让它们可以对高层协议栈进行操作。首先，设备驱动程序可能会通过调用 `register_netdevice` 或 `unregister_netdevice` 在内核中进行注册或注销。调用者首先填写 `net_device` 结构，然后传递这个结构进行注册。内核调用它的 `init` 函数（如果定义了这种函数），然后执行一组健全性检查，并将新设备添加到设备列表中（内核中的活动设备链表）。

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



设备无关接口



要从协议层向设备发送数据，需要使用 **dev_queue_xmit** 函数，这个函数对数据进行排队，并交由底层设备驱动程序进行最终传输报文的接收通常是使用 **netif_rx** 执行的。当底层设备驱动程序接收到一个报文（包含在所分配的 **sk_buff** 中）时，就会通过调用 **netif_rx** 将数据上传至设备无关层，然后，这个函数通过 **netif_rx_schedule** 将 **sk_buff** 在上层协议队列中进行排队，供以后进行处理。

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



驱动程序



www.enjoylinux.cn

网络体系结构的最底部是
负责管理物理网络设备的
设备驱动程序层。



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



Contents



Linux网络体系架构

网卡驱动设计

Dm9000网卡驱动分析

Linux输入子系统

触摸屏驱动设计

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



每个网络接口都由一个**net_device**结构来描述，
该结构可使用如下内核函数动态分配：

- 1、`struct net_device *alloc_netdev(int sizeof_priv, const char *mask, void (*setup)(struct net_device *))`
`sizeof_priv` 私有数据区大小； `mask`: 设备名； `setup` 初始化函数
- 2、`struct net_device *alloc_etherdev(int sizeof_priv)`

两个函数的关系？

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



net_device



结构 net_device 的主要成员包括:

- ✓ char **name**[IFNAMSIZ]
设备名,如: eth%d
- ✓ unsigned long **state**
设备状态
- ✓ unsigned long **base_addr**
I/O 基地址
- ✓ unsigned int **irq**
中断号

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



net_device



www.enjoylinux.cn

✓ `int (*init)(struct net_device *dev)`

初始化函数。该函数在 `register_netdev` 时被调用来完成对 `net_device` 结构的初始化

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



net_device



www.enjoylinux.cn

和字符驱动一样,网络设备也要声明能操作它的函数。有些操作可以保留为 **NULL**,有的可以通过 **ether_setup** 来使用默认设置。网络接口的设备方法可分为两组:基本的和可选的,基本方法包括那些使用接口所必需的;可选的方法实现更多高级的功能。

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



基本方法



www.enjoylinux.cn

✓ **int (*open)(struct net_device *dev)**

打开接口。ifconfig 激活时，接口将被打开。

✓ **int (*stop)(struct net_device *dev)**

停止接口。该什么时候调用呢？

✓ **int (*hard_start_xmit) (struct sk_buff *skb, struct net_device *dev)**

数据发送函数。

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



可选操作



✓ `int (*do_ioctl)(struct net_device *dev, struct ifreq *ifr, int cmd)`

处理特定于接口的 **ioctl** 命令

✓ `int (*set_mac_address)(struct net_device *dev, void *addr)`

改变**Mac**地址的函数,需要硬件支持该功能

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



www.enjoylinux.cn

网络接口驱动的注册方式与字符驱动不同之处在于**它没有主次设备号**，并使用如下函数注册。

```
int register_netdev(struct net_device *dev)
```

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



sk_buff



www.enjoylinux.cn

Linux内核中的每个网络数据包都由一个套接字缓冲区结构 **struct sk_buff** 描述，即一个 **sk_buff** 结构就是一个包，指向 **sk_buff** 的指针通常被称做 **skb**。

该结构包含如下重要成员：

- **struct device *dev;** //处理该包的设备
- **__u32 saddr;** //IP源地址
- **__u32 daddr;** //IP目的地址
- **__u32 raddr;** //IP路由器地址

嵌入式Linux技术咨询QQ号：550491596

嵌入式Linux学习交流QQ群：65212116



sk_buff



www.enjoylinux.cn

```
# unsigned char *head;    //分配空间的开始
# unsigned char *data;    //有效数据的开始
# unsigned char *tail;    //有效数据的结束
# unsigned char *end;     //分配空间的结束
# unsigned long len;      //有效数据的长度
```

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



Skb操作函数



www.enjoylinux.cn

操作sk_buff的内核函数如下:

- `struct sk_buff *alloc_skb(unsigned int len, int priority)`
分配一个sk_buff 结构, 供协议栈代码使用
- `struct sk_buff *dev_alloc_skb(unsigned int len)`
分配一个sk_buff 结构, 供驱动代码使用

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



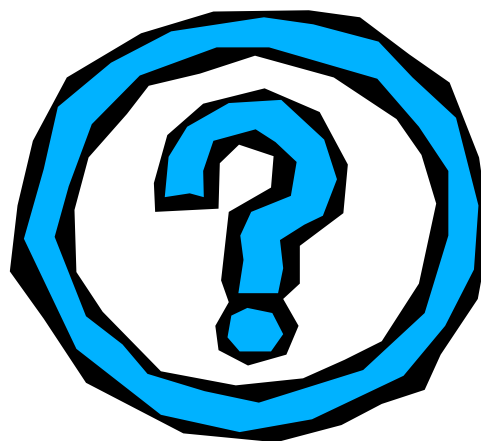
Skb操作函数



www.enjoylinux.cn

`unsigned char *skb_push(struct sk_buff *skb, int len)`

向后移动**skb**的**tail**指针，并返回**tail**移动之
前的值。函数常用来：



嵌入式Linux技术咨询QQ号：550491596

嵌入式Linux学习交流QQ群：65212116





www.enjoylinux.cn

Skb操作函数

`unsigned char *skb_put(struct sk_buff *skb, int len)`

向前移动**skb**的**head**指针，并返回**head**移动**之后**的值。函数常用来：



嵌入式Linux技术咨询QQ号：550491596

嵌入式Linux学习交流QQ群：65212116



Skb操作函数



www.enjoylinux.cn

✓ kfree_skb(struct sk_buff *skb)

释放一个sk_buff 结构，供协议栈代码使用

✓ dev_kfree_skb(struct sk_buff *skb)

释放一个sk_buff 结构，供驱动代码使用

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



设备打开



Open 请求任何它需要的系统资源并且启动接口：

- ✓ 注册中断，DMA等
- ✓ 设置寄存器，启动设备
- ✓ 启动发送队列

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



设备打开-例



www.enjoylinux.cn

```
int net_open(struct net_device *dev)
{
    /*申请中断*/
    request_irq(dev->irq, &net_interrupt, SA_SHIRQ,
        "dm9000", dev);

    /* 设置寄存器,启动设备 */
    .....

    /*启动发送队列*/
    netif_start_queue(dev);
}
```

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



数据发送



当核心需要发送一个数据包时，它调用
hard_start_transmit函数，该函数将最终调
用到**net_device**结构中的**hard_start_xmit**函
数指针。

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



数据接收



www.enjoylinux.cn

网络接口驱动可以实现两种方式的报文接收：
中断和查询，Linux中驱动多采用中断方式。



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



接收流程



1、分配Skb

skb = dev_alloc_skb(pkt->datalen + 2)

2、从硬件中读取数据到Skb

3、调用netif_rx将数据交给协议栈

netif_rx(skb)

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



中断处理



网络接口通常支持3种类型的中断：**新报文到达中断**、**报文发送完成中断**和**出错中断**。中断处理程序可通过查看网卡中的中断状态寄存器，来分辨出中断类型。

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



Contents



Linux网络体系架构

网卡驱动设计

Dm9000网卡驱动分析

Linux输入子系统

触摸屏驱动设计

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



芯片介绍



DM9000是开发板经常采用的网络芯片，是一种高度集成而且功耗很低的高速网络控制器，可以和**CPU**直连，支持**10/100M**以太网连接，芯片内部自带**16K**的**SRAM**（**3KB**用来发送，**13KB**用来接收）。

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



芯片介绍



Dm9000在收到由上层发来的以太网帧后,开始侦听网络线路,如果线路忙,就等到线路空闲为止,否则立即发送该数据帧。接收时,它将从以太网收到的数据包在经过解码、去掉帧头和地址检验等步骤后缓存在片内。在**CRC**校验通过后,它会通知**CPU**收到了数据帧。

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



参考 《DM9000.pdf》

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



工作原理



参考 《DM9000.pdf》

9.1 host interface

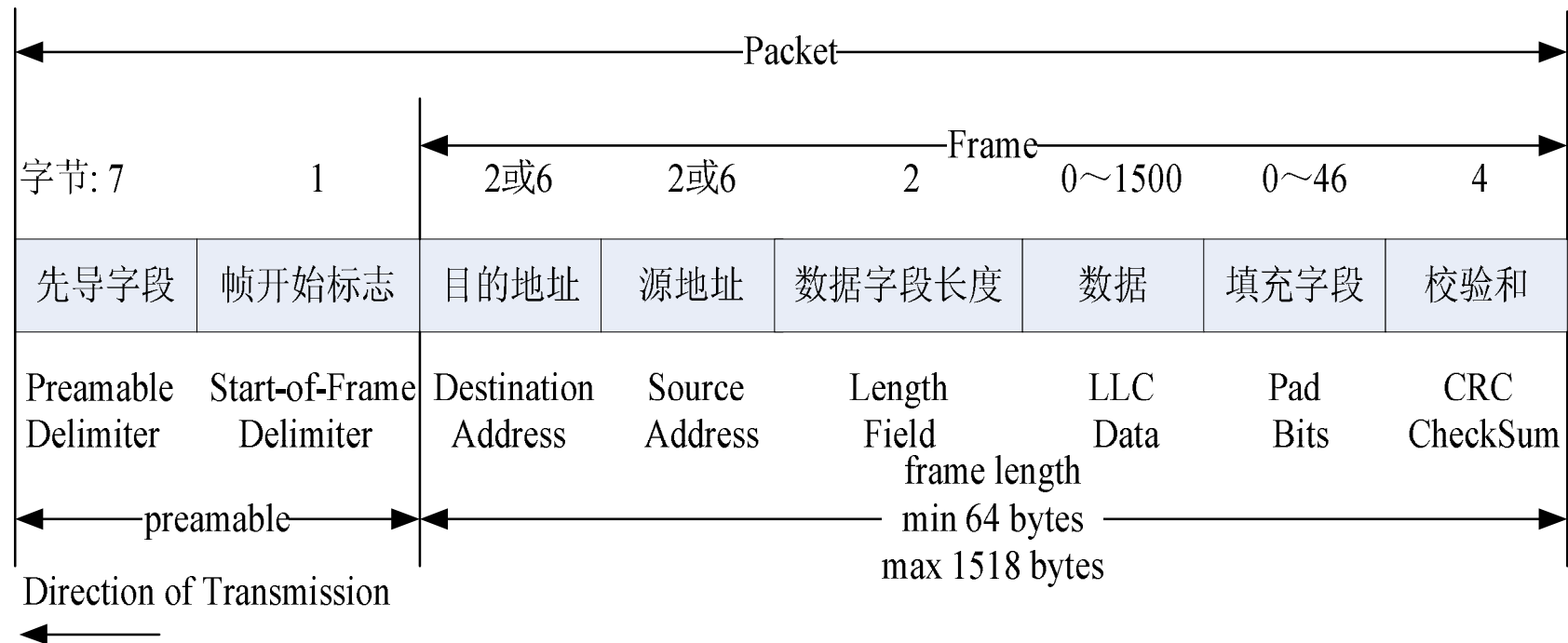
9.3 packet transmission

9.4 packet reception

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



以太网帧



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



dm9000.c



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



实验一



www.enjoylinux.cn

dm9000网卡驱动程序设计

要求:

- 1、根据注释，完成网卡驱动程序
- 2、在用户空间通过命令控制网卡

(注：不要将网卡驱动编译进内核，采用ramdisk启动，网卡驱动在内核启动后作为模块加载)

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



实验二



回环网卡驱动程序设计

要求:

实现回环网卡的发送、接收功能

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



Contents



www.enjoylinux.cn



Linux网络体系架构

网卡驱动设计

Dm9000网卡驱动分析

Linux输入子系统

触摸屏驱动设计

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



输入子系统

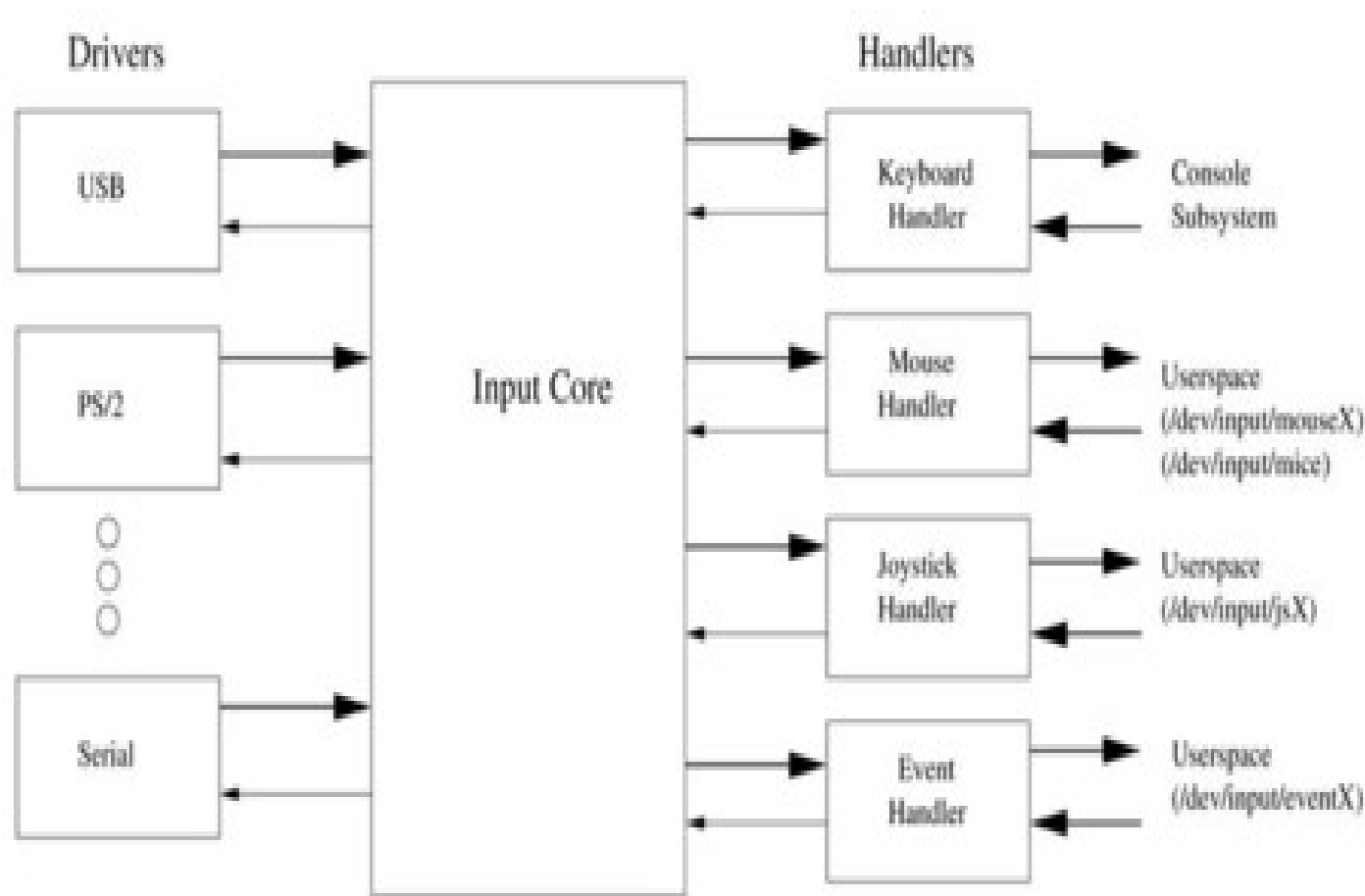


前面章节讲解按键设备驱动，实际上，在Linux系统中，一种更值得推荐的实现这类设备驱动的方法是利用**input子系统**。Linux系统提供了input子系统，按键、触摸屏、鼠标等都可以利用input接口函数来实现设备驱动。

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



体系结构



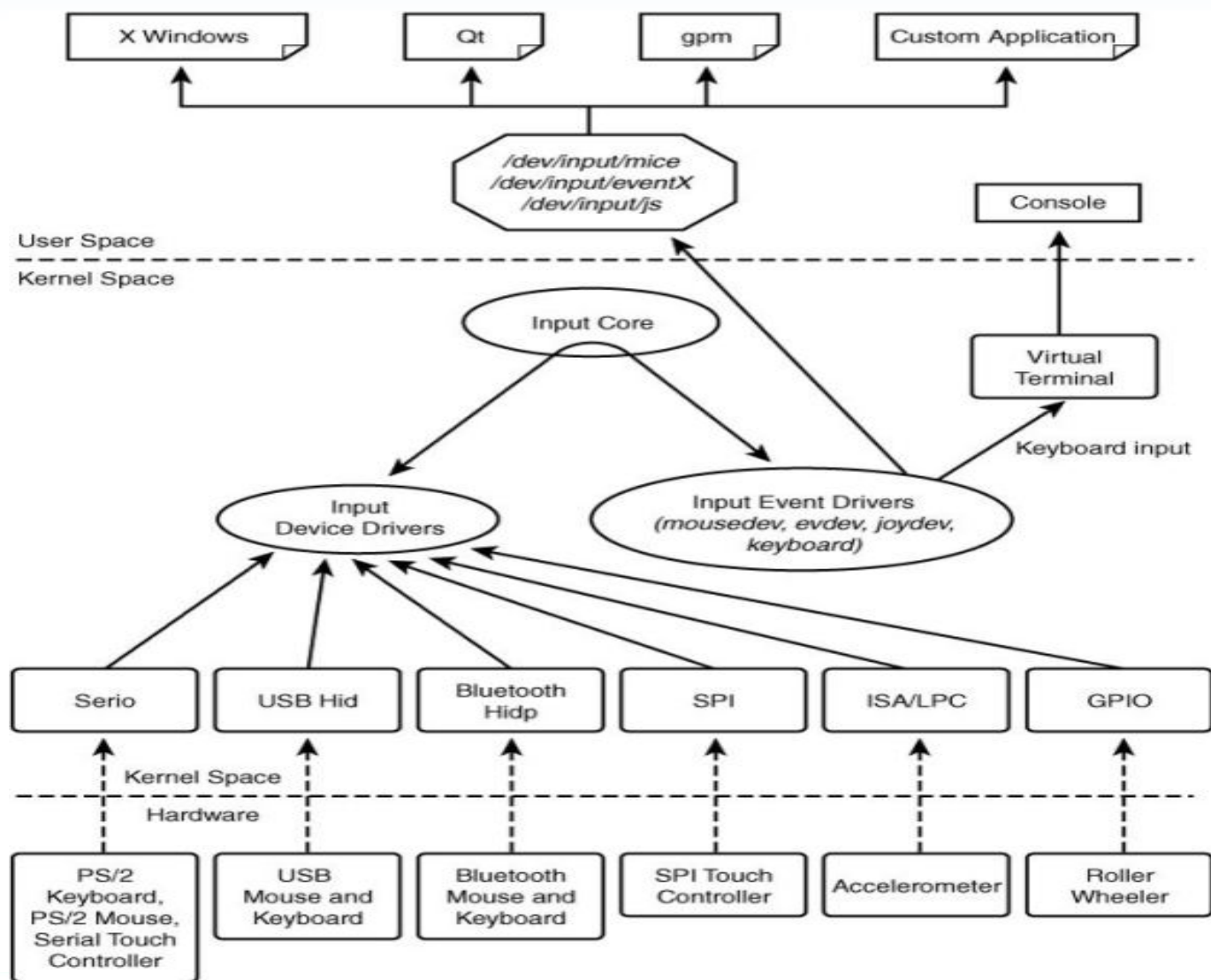
嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



体系结构



www.enjoylinux.cn



嵌入式Linux学习交流QQ群: 65212116



设备描述



在Linux内核中，input设备用**input_dev**结构体描述，使用input子系统实现输入设备驱动的时候，驱动的核心工作是向系统报告按键、触摸屏、键盘、鼠标等输入事件（**event**，通过**input_event**结构体描述），不再需要关心文件操作接口，因为input子系统已经完成了文件操作接口。驱动报告的事件经过InputCore和 Eventhandler最终到达用户空间。

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/注销



✓ 注册输入设备的函数为：

```
int input_register_device(struct input_dev *dev)
```

✓ 注销输入设备的函数为：

```
void input_unregister_device(struct input_dev *dev)
```

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



驱动实现-事件支持



设备驱动通过set_bit()告诉input子系统它支持哪些事件，如下所示：

`set_bit(EV_KEY, button_dev.evbit)`

Struct `input_dev`中有两个成员，一个是evbit；一个是keybit。分别用来表示设备所支持的事件类型和按键类型。

事件类型：

EV_RST Reset

EV_REL 相对坐标

EV_MSC 其它

EV_SND 声音

EV_FF 力反馈

EV_KEY 按键

EV_ABS 绝对坐标

EV_LED LED

EV_REP Repeat

嵌入式Linux技术咨询QQ号：550491596

嵌入式Linux学习交流QQ群：65212116



驱动实现-报告事件



用于报告EV_KEY、EV_REL和EV_ABS事件的函数分别为：

- ✓ void input_report_key(struct input_dev *dev, unsigned int code, int value)
- ✓ void input_report_rel(struct input_dev *dev, unsigned int code, int value)
- ✓ void input_report_abs(struct input_dev *dev, unsigned int code, int value)

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



驱动实现-报告事件



code:

事件的代码。如果事件的类型是EV_KEY, 该代码code为设备键盘代码。代码值0~127为键盘上的按键代码, 0x110~0x116 为鼠标上按键代码, 其中0x110(BTN_LEFT)为鼠标左键, 0x111(BTN_RIGHT)为鼠标右键, 0x112(BTN_MIDDLE)为鼠标中键。其它代码含义请参看include/linux/input.h文件

value:

事件的值。如果事件的类型是EV_KEY, 当按键按下时值为1, 松开时值为0。

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



驱动实现-报告事件



input_sync()用于事件同步，它告知事件的接收者：驱动已经发出了一个完整的报告。

例如，在触摸屏设备驱动中，一次坐标及按下状态的整个报告过程如下：

- ✓ `input_report_abs(input_dev, ABS_X, x);` //X坐标
- ✓ `input_report_abs(input_dev, ABS_Y, y);` //Y坐标
- ✓ `input_report_abs(input_dev, ABS_PRESSURE, pres);`
//压力
- ✓ `input_sync(input_dev);` //同步

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

*/*在按键中断中报告事件*/*

```
static void button_interrupt(int irq, void *dummy, struct pt_regs *fp)
{
    input_report_key(&button_dev, BTN_0, inb(BUTTON_PORT0));
    input_report_key(&button_dev, BTN_0, inb(BUTTON_PORT1));
    input_sync(&button_dev);
}
```

```
static int __init button_init(void){
```

*/*申请中断*/*

```
if (request_irq(BUTTON_IRQ, button_interrupt, 0, "button", NULL))
    return -EBUSY;
```

```
set_bit(EV_KEY, button_dev.evbit)//支持EV_KEY事件
set_bit(BTN_0, button_dev.keybit); //设备支持两个键
set_bit(BTN_1, button_dev.keybit);
input_register_device(&button_dev); //注册input设备
```

```
}
```

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



应用程序



www.enjoylinux.cn

```
struct input_event
{
    struct timeval time; //按键时间
    __u16 type; //类型, 在下面有定义
    __u16 code; //要模拟成什么按键
    __s32 value; //是按下还是释放
}

struct input_event ev_mouse[2];
fd = open ("/dev/input/event3", O_RDWR);
while(1)
{
    count=read(fd, ev_mouse, sizeof(struct input_event));
    for(i=0;i<(int)count/sizeof(struct input_event);i++)
    {
        if(EV_REL==ev_mouse[i].type)
        {
            printf("time:%ld.%d",ev_mouse[i].time.tv_sec,ev_mouse[i].time.tv_usec);
            printf(" type:%d code:%d value:%d\n",ev_mouse[i].type,ev_mouse[i].code,ev_mouse[i].value);
        }
        if(EV_KEY==ev_mouse[i].type)
        {
            printf("time:%ld.%d",ev_mouse[i].time.tv_sec,ev_mouse[i].time.tv_usec);
            printf(" type:%d code:%d value:%d\n",ev_mouse[i].type,ev_mouse[i].code,ev_mouse[i].value);
        }
    }
}
```

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

1.将按键驱动程序按照input驱动模式改写

2.改写相应的按键访问应用程序

(在mini2440平台实现)

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



Contents



Linux网络体系架构

网卡驱动设计

Dm9000网卡驱动分析

触摸屏驱动设计

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



原理概述



www.enjoylinux.cn

触摸屏分为电阻式、电容式、声表面波式和红外线扫描式等类型，使用得最多的是4线电阻式触摸屏。 S3C2440触摸屏由横向电阻比和纵向电阻线组成，由nYPON、YMON、nXPON、XMON四个控制信号控制4个MOS管（S1、S2、S3、S4）的通断。

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



控制器



S3C2440触摸屏控制器有2种处理模式：

- ① **X/Y位置分别转换模式。** 触摸屏控制器包括两个控制阶段，X坐标转换阶段和Y坐标转换阶段。
- ② **X/Y位置自动转换模式。** 触摸屏控制器将自动转换X和Y坐标。

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



工作流程



1. **Select Separate X/Y Position Conversion Mode or Auto (Sequential) X/Y Position Conversion Mode to get X/Y position.**
2. **Set Touch Screen Interface to Waiting Interrupt Mode,**
3. **If interrupt occurs, then appropriate conversion (Separate X/Y Position Conversion Mode or Auto (Sequential) X/Y Position Conversion Mode) is activated.**
4. **After get the proper value about X/Y position, return to Waiting for Interrupt Mode.**

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



中断等待



When Touch Screen Controller is in Waiting for Interrupt Mode, it waits for Stylus down. The controller, generates Interrupt (INT_TC**) signals when the Stylus is down on Touch Screen Panel. After an interrupt occurs, X and Y position can be read by the proper conversion mode (Separate X/Y position conversion Mode or Auto X/Y Position Conversion Mode).**

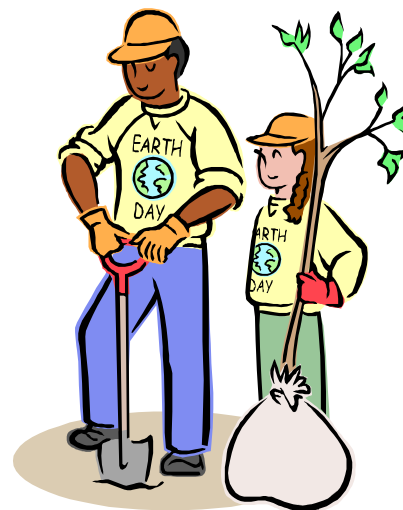
嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



实例分析



触摸屏驱动程序



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



实验



触摸屏驱动程序设计

要求:

- 1、根据注释，完成驱动程序
- 2、修改驱动，校准触摸屏

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

