

基于 UML 的嵌入式系统设备驱动分析技术

马亮, 李莉, 沈苏彬

(南京邮电大学 计算机科学与技术学院, 江苏 南京 210003)

摘要: 提出一种使用 UML(统一建模语言)对 Linux 设备驱动进行分析的方法, 借助 UML 对设备驱动程序模型进行建模并分析, 从而明晰设备驱动组件之间的依赖关系, 明确设备驱动开发的重点。在此基础上对 SDWiFi 设备驱动程序模型进行了研究, 设计处理器与 Wi-Fi 芯片之间的 SDIO 通信接口, 移植 SDWiFi 设备驱动程序。最后使用无线工具包对 SDWiFi 无线网卡设备进行了配置和测试, 测试结果表明 SDWiFi 无线网卡设备工作正常, 通过使用 UML 对设备驱动程序模型进行建模和分析, 可以提高设备驱动程序开发效率。

关键词: 驱动程序模型; 统一建模语言; 嵌入式系统

中图分类号: TP312

文献标识码: A

文章编号: 1000-436X(2012)Z2-0222-06

Technique of analyzing device drivers of embedded linux system using UML

MA Liang, LI Li, SHEN Su-bin

(School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: A method to model and analyze the device drivers in embedded Linux system using UML was proposed. By using UML modeling method, the dependencies among the device driver components and the focus of the device driver development are clarified. On this basis, the SDWiFi device driver model was studied, the SDIO communication interface between the host and Wi-Fi chips was designed and the SDWiFi device driver was transplanted. At last, the SDWiFi wireless LAN equipment was configured and tested using wireless tools. The test results show that the SDWiFi wireless LAN equipment works properly. The efficiency of the device driver development is improved by modeling and analyzing the device drivers using UML.

Key words: driver model; unified modeling language; embedded system

1 引言

统一建模语言 UML 是由单一元模型支持的一组图示方法^[1], 包括活动图、类图、构件图等。这些图示方法有助于清晰地表述和设计软件系统, 特别是采用面向对象(OO)方法构建的软件系统。UML2.0 的推出极大地推动了 MDA(model-driven architecture) 实时(real-time)系统开发模式, 以强化嵌入式实时

系统开发的设计与管理, 提升系统的质量^[2]。

嵌入式系统多采用 Linux 内核, Linux 内核完全是由 C 语言和汇编写成^[3], 但是其中用到很多面向对象的设计思想。在设备驱动方面, Linux 对同类设备采用框架设计, 框架中的核心层实现了设备通用的功能, 具体设备可以在框架的基础上进行重载, 在包含框架通用功能的同时, 设计具体设备私有的功能。另外, 一个 Linux 设备驱动程序往往被

收稿日期: 2012-07-25

基金项目: 江苏省科技支撑计划基金资助项目(BE2009157)

Foundation Item: The Scientific and Technological Funding Projects of Jiangsu Province(BE2009157)

划分为若干组件，这些组件可以是设备驱动程序中的一个模块，或者是其他驱动程序，因此 Linux 设备驱动程序的组件之间存在着一定的依赖性。组件之间的依赖，决定了整个设备驱动程序的稳定性和质量。因此，在设备驱动开发中，有必要对驱动程序各个组件之间的依赖关系进行清楚的分析。在 Linux 设备驱动开发过程中，使用 UML 对设备驱动模型进行建模和分析，有助于驱动程序开发人员理解设备驱动的组成以及各部分的交互关系，分析清楚设备驱动开发工作的重点难点。

本文阐述了一种使用 UML 对 Linux 设备驱动进行分析的方法，在此基础上，对 SDWiFi 设备驱动程序的模型进行了研究，对 SDWiFi 设备驱动进行了移植，并使用无线工具包对无线网卡设备进行了配置测试和验证。

2 设备驱动分析的 UML 方法

UML 的设计目标之一是为开发团队提供标准通用的设计语言来开发和构建计算机应用，集成了面向对象软件工程的思想^[4]。嵌入式系统设备驱动模型复杂，涉及较多组件，组件之间的依赖关系繁杂，在实际应用中，尚未形成统一的分析方法，因此本文提出一种基于 UML 的嵌入式系统设备驱动分析方法，使用 UML 建立嵌入式系统设备驱动模型。在 Linux 嵌入式实时系统的开发中，UML 可以作为高效的建模语言应用在实时系统程序中。将嵌入式系统中的对象理解为嵌入式对象^[5]，通过类似概念的转换，使用 UML 对嵌入式系统中的嵌入式对象进行建模，可以帮助开发人员忽略底层硬件实现细节，从而专注于嵌入式系统的开发和设计。

Linux 操作系统将所有的设备都看成文件，以操作文件的方式对设备进行操作。对于一个具体设备，文件操作和设备驱动是同一事物的不同层次，而不是相互独立或平行的 2 个概念。可以将一个系统划分成应用、文件系统以及设备驱动 3 个层次^[6]。在 Linux2.6 版本的内核中，建立了一个统一的设备模型，使用独立的数据结构让内核获得系统整体配合的信息^[7]。Linux 内核中对设备驱动进行了抽象，该抽象可以支持不同的任务，包括各种设备类型和热插拔设备等。在 Linux2.6 设备驱动模型中，关注总线、设备和驱动 3 个实体，总线将设备和驱动进行绑定^[8]。Linux 的设备模型可以使用 UML 静态结构图进行分析，对系统中的总线、设备和类建立对

象，总线对象描述系统总线，并跟踪连接到每个总线的设备，设备对象用来描述具体设备与系统之间的连接，类对象关注设备提供的功能。

Linux 操作系统中的应用程序不直接与底层硬件进行交互，操作系统对设备对象直接进行操作，设备对象的驱动程序完成操作系统操作命令的解析并通过下层驱动程序与底层硬件资源交互，底层硬件资源根据命令完成系统功能的实现。Linux 操作系统与设备驱动程序组件之间的交互关系可以使用协作图进行分析，建立操作系统、底层硬件资源和中间驱动程序之间的协作关系，构建设备功能实现过程。

总线是处理器与一个或者多个设备之间的通道，在 Linux 设备模型中，设备都是通过总线连接。系统在对总线注册设备的时候，会查找对应的驱动，对应的，系统在注册驱动的时候会查找对应的设备，总线完成对设备和驱动的匹配过程。Linux 设备驱动程序核心以及具体驱动程序之间的交互可以使用 UML 序列图进行分析，建立设备驱动的注册流程以及注册过程中各个驱动组件之间的交互内容和顺序过程。

通过 UML 对设备驱动程序进行建模，在建立设备驱动程序的结构模型、协作模型以及交互模型之后，能够将设备驱动组件之间复杂的关系表示清楚，能够从中明晰设备驱动程序组件之间的依赖关系，明确设备驱动程序开发工作的重点。

3 SDWiFi 设备驱动

SDWiFi 设备驱动程序包括 SDIO 驱动和 88W8686 驱动 2 个主要部分。在 SDWiFi 设备驱动中，SDIO 驱动和 88W8686 驱动之间存在很强的依赖性，其中的调用关系错综复杂。本文使用 UML 对 SDWiFi 设备驱动模型进行建模，对 SDWiFi 设备驱动建立静态结构图、协作图和序列图，细化 SDWiFi 驱动程序的结构，理清驱动程序移植的工作重点。

3.1 系统平台

系统硬件平台由 TMS320DM355 和 WM-G-MR-09 2 个部分组成，TMS320DM355 部分包括系统 CPU 芯片 TMS320DM355 及其外围电路，包括 DDR、NAND 等构成基本最小系统。WM-G-MR-09 部分包括 WM-G-MR-09 模组及其外围电路，包括射频天线等。TMS320DM355 通过 SDIO 接口与 WM-G-MR-09 相连。TMS320DM355 与 WM-G-MR-09 之间的信息交互，全部通过 SDIO 接口实现，SDWiFi

驱动移植工作的重点就在于 SDIO 接口的设计以及通过 SDIO 的数据通信。

系统软件包括操作系统和 SDWiFi 设备驱动程序 2 个部分。操作系统采用 TI 随 TMS320DM355 平台提供的 Linux 操作系统，该操作系统基于 Linux2.6.32 内核。SDWiFi 驱动程序包括 2 个主要部分，SDIO 驱动和 88W8686 驱动，Linux 内核代码目录/drivers/mmc 下包含对 SDIO 的驱动程序实现，目录/drivers/net/wireless/libertas 下包含对 88W8686 的驱动程序实现。

3.2 88W8686 软件结构

Marvell 88W8686 解决方案提供了 Wi-Fi 的软件架构，包括固件和主机驱动程序 2 个部分^[9]，如图 1 所示。固件程序是 88W8686 的基本控制程序，负责 IEEE 802.11 提供的服务，包括认证、解除认证、登录等。固件程序可执行文件由 Marvell 提供，通过控制 88W8686 芯片中的 CPU、媒体接入控制层、基带处理和射频发送接受模块，实现网络数据的无线收发以及 IEEE 802.11 相关服务功能。固件程序向主机提供了应用程序接口，主机通过 88W8686 与主机之间的 SDIO/G-SPI 接口，与固件程序交互，获得 88W8686 芯片的控制，完成无线网卡功能，将上层数据信息封装成帧并通过无线网卡发送出去，或者将无线网卡收到的信息进行解封装，将数据提供给上层应用程序。本文中，根据 TMS320DM355 开发板情况，选用 SDIO 模式接口与 88W8686 交互，在 88W8686 驱动加载过程中，通过 SDIO 接口将 88W8686 所需的固件程序下载到 88W8686 芯片中，固件程序加载完成之后，TMS320DM355 便可以通过固件程序与 88W8686 交互，控制芯片资源，实现无线网卡功能。

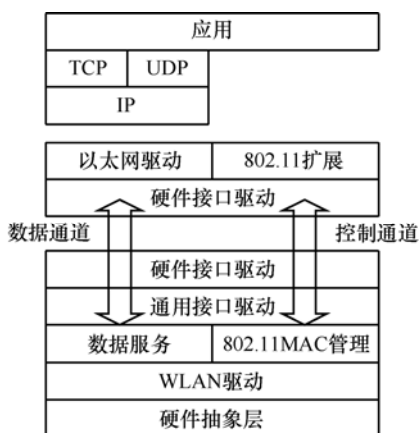


图 1 主机驱动程序与固件程序的软件架构

3.3 SDWiFi 设备驱动模型

在 SDIO 驱动程序中，定义了 SDIO 总线类型，定义了 SDIO 总线的设备驱动，对设备驱动进行了封装。定义了 SDIO 总线的设备驱动注册函数，对 SDIO 总线驱动注册函数进行了封装，为 88W8686 的总线驱动注册提供了注册函数的接口，通过对 SDIO 总线驱动清除函数的封装，提供了总线驱动清除函数接口。定义了 SDIO 总线类型设备，定义了 SDIO 总线类型设备的注册和清除函数。Linux 中 SDIO 和 MMC 的命令接口相互兼容，因此 Linux 对 MMC 的操作进行了封装，用于描述 CPU 端 MMC 的接口设备，将底层的实现细节抽象成类，而接入的 SDIO 设备，例如 SD 卡，则会被封装成数据结构，通过向该数据结构发送命令进行相关操作。MMC 在 SDIO 总线之外又定义了一个 MMC 总线，以及相应的设备结构，定义相应的类用于存储块设备的操作参数，以实现 SD 卡的块存储设备操作。对于 SDIO 则不需要该 MMC 总线。通过以上分析，可以建立包含 SDWiFi 在内的设备结构模型。SDWiFi 设备驱动的静态结构如图 2 所示。

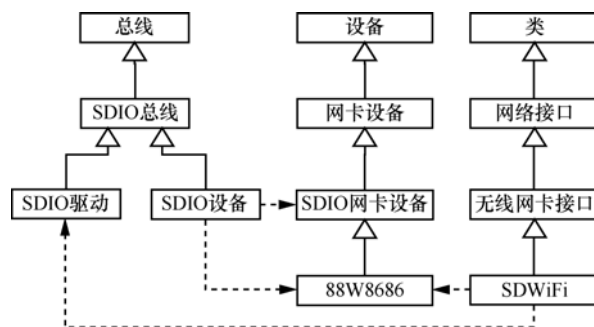


图 2 SDWiFi 设备驱动静态结构

从 SDWiFi 设备驱动静态结构中清晰的看出，总线树跟踪了连接到每个总线的设备，SDIO 总线跟踪了连接到 SDIO 总线上设备，包括 SDIO 设备。在 SDWiFi 设备驱动模型中，这里的 SDIO 设备指向 SDIO 网卡设备，也即 88W8686 芯片。网络接口类则管理了无线网卡接口 SDWiFi，为应用层提供网络接口功能。SDWiFi 无线网卡接口功能的实现依赖于 88W8686 设备以及 SDIO 驱动。

应用程序忽略 SDWiFi 的底层实现细节，直接与 SDWiFi 进行交互，将 SDWiFi 设备作为网卡进行操作。SDWiFi 设备对上层操作进行解析，并通过 SDIO 驱动将解析后的操作命令发送给 88W8686，88W8686 的固件程序根据接收到的命令

实现相应的功能，调用 88W8686 硬件资源，实现数据的无线收发。通过以上分析，可以建立从应用程序到硬件设备之间的协作模型。SDWiFi 设备驱动的协作图如图 3 所示。



图 3 SDWiFi 设备驱动协作

从 SDWiFi 设备驱动协作图中可以清晰的看出，操作系统不直接操作 88W8686 芯片。操作系统的直接调用接口是 SDWiFi 无线网卡接口，SDWiFi 无线网卡接口通过 SDIO 通道与 88W8686 芯片交互。SDWiFi 无线网卡接口将操作系统提供的数据经由 SDIO 通道传送给 88W8686 芯片并由其完成数据的收发。

88W8686 网络设备及其设备驱动被挂载在 SDIO 总线上。在 88W8686 驱动中，定义了 SDIO 总线驱动结构，在驱动模块初始化函数中，通过调用 SDIO 提供的总线注册函数接口，将 88W8686 的 SDIO 总线驱动结构注册挂载到 SDIO 总线上，在模块清除函数中，通过调用 SDIO 提供的总线清除函数接口，清除 SDIO 总线驱动结构。在注册 88W8686 的 SDIO 总线设备驱动的过程中，调用 SDIO 总线上的匹配函数对 SDIO 总线设备，即 88W8686 设备进行匹配，完成匹配后将 88W8686 网络设备以及设备驱动注册挂载到 SDIO 总线上。对 88W8686 的设备操作进行了封装，将 88W8686 网络设备抽象成一个类，在对网络设备进行操作时，通过这个类发送命令和数据实现。完成 88W8686 网络设备 SDIO 总线的注册后，会调用 88W8686 网络设备驱动的探测函数，完成驱动的注册操作。在探测函数中，通过使用 uevent 通讯机制实现了用户态和内核态数据的交互，将用户态的固件数据复制到内核态，通过 SDIO 接口，将固件数据写入 88W8686 设备，完成 88W8686 设备的固件初始化。在 88W8686 通过下载的固件程序启动运行，设置运行参数和运行环境，之后主机设备可以通过 SDIO 接口与 88W8686 的固件程序交互，完成无线网卡相关功

能。通过以上分析，可以建立 SDWiFi 组件之间的交互模型。SDWiFi 设备驱动的序列图如图 4 所示。

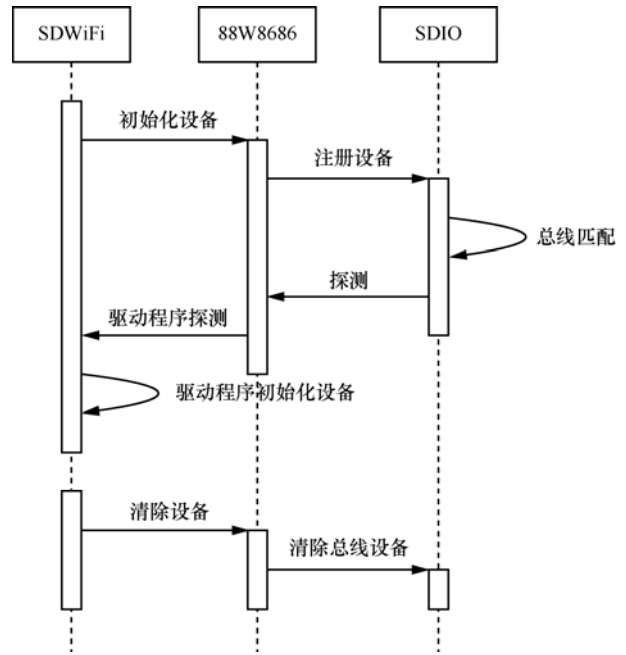


图 4 SDWiFi 设备驱动序列

从 SDWiFi 设备驱动序列图中可以清晰的看出，驱动程序的正确加载有赖于 SDWiFi 设备驱动组件，即 88W8686 驱动以及 SDIO 驱动之间注册信息的传递。因此，在 88W8686 驱动以及 SDIO 驱动程序完成的条件下，驱动移植的工作重点在于如何实现这 2 个驱动组件信息之间的交互和传递。

通过 UML 对 SDWiFi 设备驱动模型进行建模，从 SDWiFi 设备驱动的 UML 静态结构图、协作图以及序列图中可以看出，SDWiFi 的设备操作依赖于 SDWiFi 设备的建立，SDWiFi 设备的注册依赖于 SDIO 设备的注册。另外 SDWiFi 设备的初始化还需要 88W8686 固件的支持。因此，SDWiFi 设备驱动的开发主要包括 SDIO 设备驱动、88W8686 设备驱动的实现，建立 88W8686 设备驱动与 SDIO 设备驱动之间的关联，以及 88W8686 固件程序的初始化。

4 SDWiFi 设备驱动的移植

Linux 内核对于 SDIO 驱动以及 88W8686 无线网卡设备驱动分别具有较好的实现，从 SDWiFi 设备驱动模型的分析中可以看出，SDWiFi 设备驱动移植工作的主要内容在于 SDIO 总线设备的定义，建立 88W8686 驱动和 SDIO 驱动之间的依赖，88W8686 固件的配置以及无线网卡配置工具的安

装使用。另外 SDIO 驱动程序中存在部分错误，需要对其进行修改。

在 88W8686 驱动程序 SDIO 总线设备定义文件 `if_sdio.c` 中，定义 SDIO 驱动类型。模块初始化函数 `if_sdio_init_module()` 会通过 `sdio_register_driver` 函数将 `if_sdio_driver` 注册挂载到 `sdio_bus_type` 总线上。

```
static struct sdio_driver if_sdio_driver = {
    .name          = "libertas_sdio",
    .id_table       = if_sdio_ids,
    .probe         = if_sdio_probe,
    .remove        = if_sdio_remove,
    .drv = {
    .pm = &if_sdio_pm_ops,
    },
};
```

将文件 `driver/mmc/core/sdio_cis.c` 第 43 行代码 `if(nr_string<4)` 修改为 `if(nr_string==0)`。SDIO 驱动程序中，对参数 `nr_string` 的判断存在错误，需要进行修改。

配置内核选项，添加对 SDIO 驱动以及 88W8686 驱动的支持。完成后保存内核配置信息，编译内核得到内核镜像文件 `uImage`。

```
Linux Kernel Configuration
[*]Networking support
[*]Wireless
<*>cfg80211 – wireless configuration API
Device Drivers
Generic Driver Options
[*]userspace firmware loading support
<*>MMC/SD/SDIO card support
[*]Network device support
[*]Wireless LAN
[*]Wireless LAN (IEEE 802.11)
[*]Marvell 8xxx Libertas WLAN driver support
[*]Marvell Libertas 8385/8686/8688 SDIO
802.11b/g cards
```

从 Marvell 网站下载 Marvell 88W8686 的固件 `helper_sd.bin` 和 `sd8686.bin`。将 `helper_sd.bin` 重命名为 `sd8686_helper.bin`，将 `sd8686_helper.bin` 和 `sd8686.bin` 2 个固件复制到根文件系统 `/lib/firmware/` 下。

5 SDWiFi 测试

5.1 无线工具包

无线工具包 (wireless tools for Linux) 是一个

Linux 命令行工具包，使用该工具包可以对支持无线扩展 (Linux wireless extension) 的无线设备进行配置。无线工具包和无线扩展由 Jean Tourrilhes 在维护，惠普 (Hewlett-Packard) 赞助^[10]。无线扩展是一组通用的应用程序接口，能在用户空间对通用无线局域网进行配置和统计。无线工具包用来操作无线扩展的工具集，使用字符界面，支持所有无线扩展，包括 `iwconfig`、`iwlist` 等工具。

下载软件包 `wireless_tools.29.tar.gz`，解压后，修改顶层 `Makefile` 文件，其中 `PREFIX` 定义安装路径，`CC` 和 `AR` 为交叉编译工具链选项。交叉编译并安装到本地，复制安装目录下 `lib` 内所有文件至目标板文件系统的 `lib` 目录下，这样，目标板上就可以使用 `iwconfig`、`iwlist` 等工具集命令配置无线网卡参数。

5.2 SDWiFi 测试结果

使用 Uboot 引导 Linux 内核镜像 `uImage` 和根文件系统 `ramdisk`，可以从串口终端看到 Linux 内核启动输出的信息。从 Linux 内核启动过程中看到如下信息，可以看到 SDWiFi 驱动已经成功加载。

```
mmc0: new SDIO card at address 0001
libertas_sdio: Libertas SDIO driver
libertas_sdio: Copyright Pierre Ossman
libertas_sdio mmc0:0001:1: firmware: requesting
sd8686_helper.bin
libertas_sdio mmc0:0001:1: firmware: requesting
sd8686.bin
libertas: e0:2a:82:71:d1:74, fw 9.70.3p24, cap
0x00000303
```

```
libertas: wlan0: Marvell WLAN 802.11 adapter
```

使用命令 `ifconfig wlan0 up` 启动 SDWiFi 无线网卡，使用命令 `iwlist wlan0 scan` 和 `iwconfig wlan0 essid "NJUPT"` 扫描并加入 AP，使用命令 `udhcpc-i wlan0` 获取 IP 和 DNS，使用命令 `iwconfig wlan0` 或 `ifconfig wlan0` 查看 SDWiFi 无线网卡状态。此时，使用 `ping` 命令测试可以发现目标板与网关连接良好。由测试结果可知，SDWiFi 无线网卡能够正常工作。

在使用 UML 对 SDWiFi 设备驱动模型进行分析，建立 SDWiFi 设备驱动的结构模型、组件之间的协作模型以及交互模型后，从中分析得出驱动移植的工作重点，简化了驱动移植的步骤，减短了驱动的开发周期，提高了工作效率。

6 结束语

嵌入式 Linux 设备驱动模型结构复杂，驱动程序的组件之间存在着依赖性，分析设备驱动模型并理清驱动程序组件之间的依赖关系，对于设备驱动程序的开发至关重要。本文阐述了一种使用 UML 对 Linux 设备驱动建模进而分析的方法，通过使用 UML 对设备模型进行建模，可以分析清楚设备驱动的层次结构和依赖关系，明确设备驱动开发的重点。在此基础上，对 SDWiFi 设备驱动模型进行了分析，建立了 SDWiFi 驱动的结构模型，协作模型以及交互模型，从中可以清楚地看出 SDWiFi 设备驱动中的组件 88W8686 和 SDIO 驱动之间的层次结构和依赖关系，明确了 SDWiFi 设备驱动开发的重点，设计处理器与 WiFi 芯片之间的 SDIO 通信接口，并移植实现 SDWiFi 设备驱动程序。使用无线工具包对 SDWiFi 无线网卡设备进行了配置和测试，测试结果表明 SDWiFi 无线网卡设备工作正常，SDWiFi 设备驱动程序移植成功。

参考文献：

- [1] FOWLER M. UML 精粹：标准对象建模语言简明指南[M]. 北京：清华大学出版社，2005.5.
FOWLER M. UML Distilled-A Brief Guide to the Standard Object Modeling Language[M]. Beijing:Tsinghua University Press,2005.5.
- [2] 高焕堂. UML 嵌入式设计[M]. 北京:清华大学出版社, 2008.1.
GAO H T. UML Embedded Design[M]. Beijing: Tsinghua University Press, 2008. 1.
- [3] WANG G P. Modeling C-based Embedded System using UML Design[J]. IEEE, 2009. 2973-2977.
- [4] UML specification[EB/OL]. <http://www.uml.org>, 2012.
- [5] VALLIUS T, RONING J. Implementation of the "embedded object" concept and an example of using it with UML[A]. Computational Intelligence in Robotics and Automation[C]. Espoo, Finland, 2005. 635-640.
- [6] 毛德操, 胡希明. Linux 内核源代码情景分析[M]. 浙江:浙江大学出版社, 2001.

版社, 2001.

MAO D C, HU X M. Linux Kernel Scenarios and Analyses[M]. Hangzhou: Zhejiang University Press, 2001.

- [7] CORBET J, RUBINI A, KROAB-HARTMAN G. Linux 设备驱动程序 [M]. 北京:中国电力出版社, 2005.11.

CORBET J, RUBINI A, KROAB-HARTMAN G. Linux Device Drivers[M]. Beijing: China Electric Power Press, 2005.11.

- [8] 宋宝华. Linux 设备驱动开发详解[M]. 北京:人民邮电出版社, 2010.11.

SONG B H. Linux Device Driver Development Explain[M].Beijing: Posts and Telecom Press, 2010.11.

- [9] Marvell. WLAN Subsystem Host Driver, Firmware, and Interface: Firmware Specification v8.0[S]. 2006.

- [10] HP. Wireless tools for Linux[EB/OL]. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html 2008.8.

作者简介：



马亮（1988-），男，江苏南京人，南京邮电大学硕士生，主要研究方向为嵌入式系统与软件。



李莉（1981-），女，江苏丹阳人，南京邮电大学博士生、讲师，主要研究方向为信息网络。



沈苏彬（1963-），男，江苏南京人，南京邮电大学研究员、博士生导师，主要研究方向为计算网络、下一代电信网以及网络安全。