21世纪高等学校嵌入式系统专业规划教材

张光建 刘政 编著

嵌入式Linux驱动程序开发实例教程

清华大学出版社

21世纪高等学校嵌入式系统专业规划教材

张光建 刘政 编著

嵌入式Linux驱动程序开发实例教程

清华大学出版社
北京

内 容 简 介

本书详细介绍了计算机的嵌入式 Linux 系统的三类驱动程序开发方法。全书共分 10 章。前面 4 章简要介绍 Linux 操作系统的操作、嵌入式 Linux 驱动开发环境的建立方法以及 Linux 驱动程序开发所需要的内核基础。第 5 章和第 6 章介绍字符驱动程序的设计方法，第 7 章和第 8 章介绍块设备驱动程序的设计方法，第 9 章和第 10 章介绍网络驱动程序的设计方法。每类驱动程序分别从数据结构、驱动架构、驱动模块设计三方面进行了介绍，每类驱动程序都配置了一个模拟驱动小实例，通过这些小实例，可以深入理解各类驱动程序的架构。每类驱动程序还配置了一个真实外部设备的驱动程序开发实例。

本书实例丰富，通俗易懂，可作为高等学校计算机科学与技术、软件工程等专业学生学习嵌入式技术的教材，也可以作为计算机相关专业学生学习操作系统的提高教材，还可作为工程技术人员设计 Linux 驱动程序的参考书。

图书在版编目(CIP)数据

嵌入式 Linux 驱动程序开发实例教程/张光建，刘政编著. —北京：清华大学出版社，2011.9
(21 世纪高等学校嵌入式系统专业规划教材)
ISBN 978-7-302-26059-2

Ⅰ. ①嵌… Ⅱ. ①张… ②刘… Ⅲ. ①Linux 操作系统－程序设计－高等学校－教材
Ⅳ. ①TP316.89

中国版本图书馆 CIP 数据核字(2011)第 131798 号

责任编辑：梁　颖　薛　阳
责任校对：李建庄
责任印制：李红英
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　　总　　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京嘉实印刷有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：12　**字　数**：295 千字
版　　次：2011 年 9 月第 1 版　　**印　　次**：2011 年 9 月第1 次印刷
印　　数：1～3000
定　　价：21.00 元

产品编号：041054-01

出版说明

嵌入式计算机技术是21世纪计算机技术两个重要发展方向之一，其应用领域相当广泛，包括工业控制、消费电子、网络通信、科学研究、军事国防、医疗卫生、航空航天等方方面面。我们今天所熟悉的电子产品几乎都可以找到嵌入式系统的影子，它从各个方面影响着我们的生活。

技术的发展和生产力的提高，离不开人才的培养。目前国内外各高等院校、职业学校和培训机构都涉足了嵌入式技术人才的培养工作，高校及其软件学院和专业的培训机构更是嵌入式领域高端人才培养的前沿阵地。国家有关部门针对专业人才需求大增的现状，也着手开发“国家级”嵌入式技术培训项目。2006年6月底，国家信息技术紧缺人才培养工程(NITE)在北京正式启动，首批设定的10个紧缺专业中，嵌入式系统设计与软件开发、软件测试等IT课程一同名列其中。嵌入式开发因其广泛的应用领域和巨大的人才缺口，其培训也被列入国家商务部门实施服务外包人才培训“千百十工程”，并对符合条件的人才培训项目予以支持。

为了进一步提高国内嵌入式系统课程的教学水平和质量，培养适应社会经济发展需要的、兼具研究能力和工程能力的高质量专业技术人次。在教育部相关教学指导委员会专家的指导和建议下，清华大学出版社与国内多所重点大学共同对我国嵌入式系统软硬件开发人才培养的课程框架和知识体系，以及实践教学内容进行了深入的研究，并在该基础上形成了“嵌入式系统教学现状分析及核心课程体系研究”、“微型计算机原理与应用技术课程群的研究”、“嵌入式Linux课程群建设报告”等多项课程体系的研究报告。

本系列教材是在课程体系的研究基础上总结、完善而成，力求充分体现科学性、先进性、工程性，突出专业核心课程的教材，兼顾具有专业教学特点的相关基础课程教材，探索具有发展潜力的选修课程教材，满足高校多层次教学的需要。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

(1) 反映嵌入式系统学科的发展和专业教育的改革，适应社会对嵌入式人才的培养需求，教材内容坚持基本理论的扎实和清晰，反映基本理论和原理的综合应用，在其基础上强调工程实践环节，并及时反映教学体系的调整和教学内容的更新。

(2) 反映教学需要，促进教学发展。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略，突出重点。规划教材建设把重点放在专业核心(基础)课程的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现工程型和应用型的专业教学内容和课程体系改革成果的教材。

(4) 支持一纲多本，合理配套。专业核心课和相关基础课的教材要配套，同一门课程可以有多本具有各自内容特点的教材。处理好教材统一性与多样化，基本教材与辅助教材、教

学参考书，文字教材与软件教材的关系，实现教材系列资源的配套。

(5) 依靠专家，择优落实。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主编。书稿完成后认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平的、以老带新的教材编写队伍才能保证教材的编写质量，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校嵌入式系统专业规划教材
联系人：魏江江 weijj@tup.tsinghua.edu.cn

前　言

计算机的嵌入式系统由硬件系统和软件系统两大部分组成。硬件系统是整个系统的物质基础，是软件运行的平台，而软件系统则实际控制系统的运行。外部设备是硬件系统中不可缺少的组成部分。驱动外部设备的软件即设备驱动程序与底层硬件直接打交道，按照硬件设备的具体工作方式读写设备寄存器，完成设备的轮询、中断处理、DMA 通信等，它充当了硬件和应用软件之间的纽带，使得应用软件只需要调用系统软件的应用编程接口就可让硬件去完成要求的工作。

在使用类似 Linux、Windows CE 操作系统的嵌入式系统中，设备驱动程序位于一个单独的层次，与内核及应用程序都有统一的接口。通常，硬件开发工程师和应用程序开发工程师都不太涉及设备驱动程序的开发，开发设备驱动程序的工作由设备驱动程序开发工程师来完成。在 Linux 中，设备驱动程序是内核的一部分，其开发所使用的知识和技术与开发应用程序所使用的知识和技术有较大差别，因此有必要专门进行讲授。

Linux 系统的设备分为字符设备、块设备和网络设备三种。要想进行基于 Linux 的设备驱动程序开发，需要具有良好的硬件基础、良好的 C 语言基础、一定的 Linux 内核基础以及良好的多任务并发控制和同步的基础。本书通过实例的形式综合运用这些基础模块。

全书共 10 章。第 1 章简述外部设备在整个嵌入式系统中的地位、设备驱动程序的作用以及学习 Linux 驱动程序开发所需要的技术基础及学习方法。第 2 章介绍 Linux 的常用命令、编辑器等，主要针对不熟悉 Linux 基本操作的学生进行介绍。第 3 章简述嵌入式 Linux 驱动开发环境的建立。第 4 章介绍三类 Linux 驱动程序开发所需要的内核基础。第 5 章、第 7 章、第 9 章分别介绍 Linux 的三类驱动程序，分别从数据结构、驱动架构、驱动模块设计三方面进行介绍。每章都结合了一个模拟驱动小实例，其特点是具有实际驱动程序完整的架构，能够编译及测试，读者通过上机调试这些小实例，能够增加对相关方法的理解。第 6 章、第 8 章、第 10 章分别针对三类 Linux 驱动程序介绍一个真实外部设备驱动程序的开发，这些章节是 Linux 驱动程序相关知识和技术的综合运用。

本书的全部实例在 UP-NETARM2410-S 实验平台上经过测试。要运行本书的实例，建议使用 CPU 为 S3C2410X、Linux 内核为 2.4 的实验平台，如北京博创、凌阳、周立功等公司的 S3C2410X 实验平台。如果要在其他实验平台上实验，则需要对实例中硬件相关部分做适度的修改。

本书针对计算机科学与技术、软件工程等计算机相关专业的学生编写的。建议总学时为 48 学时，其中理论 16 学时，实验 32 学时。

本书由重庆理工大学的张光建副教授和刘政副教授编著。由于编者水平有限，书中难免存在不够完善甚至错误之处，望读者批评指正。

作　者

2011 年 7 月

目　录

第 1 章　设备驱动概述

本章首先从嵌入式系统的硬件及软件组成开始，讲述了设备驱动程序在嵌入式系统中的作用，其次概述了 Linux 设备驱动程序的分类、特点及学习方法。

1.1　设备驱动程序的作用

1.1.1　嵌入式系统的硬件组成

与普通的计算机系统一样，嵌入式系统也是由硬件和软件两大部分组成的。前者是整个系统的物质基础，提供软件运行平台和通信（包括人机交互）接口；后者实际控制系统的运行。

嵌入式系统的硬件包括三个组成部分：处理器核、外围电路以及外设与扩展，如图 1-1 所示。

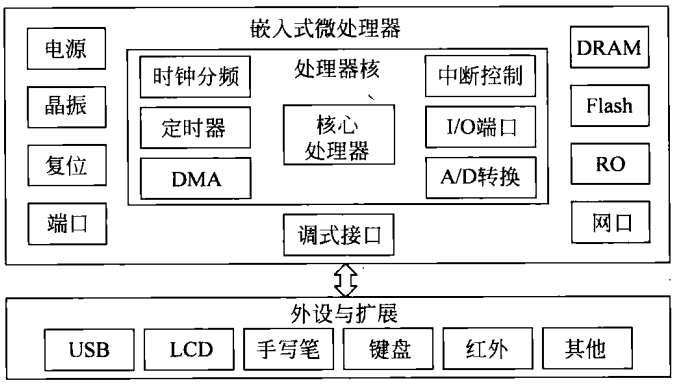

图 1-1　嵌入式系统硬件结构

1. 处理器核

嵌入式系统的核心部件，包括负责控制整个嵌入式系统执行的核心处理器、时钟分频、定时器、中断控制、I/O 端口等，也可能直接包含 A/D 和 D/A 转换处理端口。

2. 外围电路

包括嵌入式系统所需要的基本存储管理，晶振、复位、电源等控制电路及接口。它们与处理器一起构成一个完整的嵌入式微处理器。对 32 位以上的微处理器，一般还有专门的调试接口（JTAG 或 BDM）。另外，也可能直接提供网络、多媒体等处理功能。

3. 外设与扩展

该部分位于嵌入式微处理器之外，是嵌入式系统与真实环境交互的接口，可看成是板上电路。它可以提供包括扩展存储（如 Flash Card）、I/O 接口（如键盘、鼠标、LCD）等设备的

控制电路，或直接使用相关的控制芯片。此外，根据实际应用的需要，还可以扩展一些专用芯片，如加密解密、现场总线（如 CAN、1553B）、移动通信（如 CDMA、GSM）等专用芯片。

实际系统中，嵌入式系统的硬件配置非常灵活。不但外设可以根据需要进行裁剪，而且嵌入式微处理器内部的模块也可以选择。

1.1.2 嵌入式系统的软件组成

嵌入式系统的软件结构可以分为 4 个层次：板级支持包（Board Support Packet，BSP）、嵌入式操作系统、应用编程接口（API）及嵌入式应用系统，如图 1-2 所示。

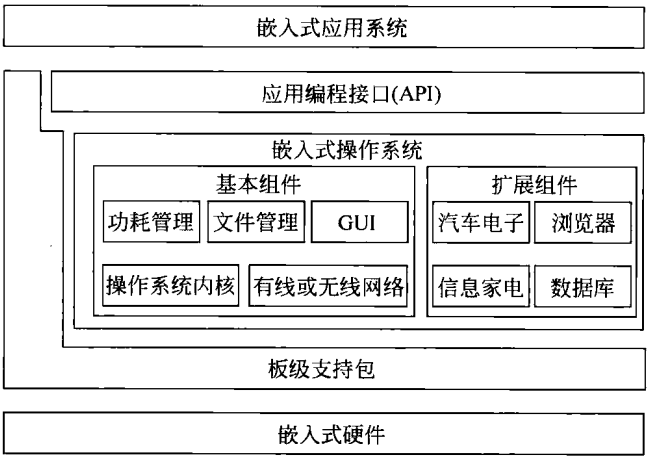

图 1-2 嵌入式系统的软件结构

1. 板级支持包

它是介于嵌入式硬件和上层软件之间的一个底层软件开发包，主要目的是屏蔽下层硬件。该层拥有两部分功能。一是系统引导，包括嵌入式微处理器和基本芯片的初始化；二是提供设备的驱动接口，负责嵌入式系统与外设的信息交互。

2. 嵌入式操作系统

这是对多任务嵌入式系统进行有效管理的核心部分，可以分成基本内核和扩展内核两部分。前者提供操作系统的核心功能，负责整个系统的任务调度、存储分配、时钟管理、中断管理，也可提供文件、GUI、网络等通用服务；后者则是根据应用领域的需要，为用户提供面向领域或面向具体行业的操作系统扩展功能，如图形处理、汽车电子、信息家电等领域的专用扩展服务。

3. 应用编程接口

也可称为嵌入式应用编程中间件，由为编制嵌入式应用程序提供的各种编程接口库（Lib）或组件（Component）组成，可以针对不同应用领域（如网络设备、PDA、机顶盒）、不同安全要求分别构建，从而减轻应用开发者的负担。

4. 嵌入式应用系统

这是最终运行在目标机上的应用软件，如嵌入式文本编辑、游戏、读写卡系统、家电控制软件、多媒体播放软件等。

1.1.3　驱动程序的作用

任何一个计算机系统的运行都是系统中软硬件协作的结果。硬件是底层基础，是所有软件得以运行的平台，代码最终会落实为硬件上的组合逻辑与时序逻辑；软件则实现了具体应用，它按照各种不同的业务需求而设计，满足了用户的需求。硬件较固定，软件则很灵活，可以适应各种复杂多变的应用。

为了尽可能快速地完成设计，应用软件工程师不想也不关心硬件，而硬件工程师也难有足够的闲暇和能力去顾及软件。例如，应用软件工程师在调用套接字发送和接收数据的时候，他不关心网卡上的中断、寄存器、存储空间、I/O 端口、片选以及其他任何硬件词汇；在使用 printf()函数输出信息的时候，他不用知道底层究竟是怎样把相应的信息输出到屏幕或串口的。因此，应用软件工程师需要看到一个没有硬件的纯粹的软件世界，硬件必须透明地呈现给他们。谁来实现硬件对应用软件工程师的隐形？这个艰巨的任务就落在了驱动工程师的头上。

对设备驱动最通俗的解释就是“驱使硬件设备行动”。设备驱动与底层硬件直接打交道，按照硬件设备的具体工作方式读写设备寄存器，完成设备的轮询、中断处理、DMA 通信，进行物理内存向虚拟内存的映射，最终使通信设备能够收发数据，使显示设备能够显示文字和画面，使存储设备能够记录文件数据。设备驱动充当了硬件和应用软件之间的纽带，它使得应用软件只需要调用系统软件的应用编程接口(API)就可让硬件去完成要求的工作。在系统中没有操作系统的情况下，工程师可以根据硬件设备的特点自行定义接口，如对串口定义 SerialSend()、SerialRecv()，对 LED 定义 LightOn()、LightOff()，以及对 FLASH 定义 FlashWrite()、FlashRead()等。而在有操作系统的情况下，设备驱动的架构则由相应的操作系统定义，驱动工程师必须按照相应的架构设计设备驱动，这样，设备驱动才能良好地整合到操作系统的内核中。

驱动程序沟通着硬件和应用软件，相应地驱动工程师沟通着硬件工程师和应用软件工程师。随着通信、电子行业的发展，全世界每天都会有大量的新芯片被生产，大量的新电路板被设计，因此，也会有大量设备驱动需要开发。这些设备驱动，或运行在简单的单任务环境中，或运行在 VxWorks、Linux、Windows 等多任务操作系统环境中，都发挥着不可替代的作用。

1.2　Linux 设备驱动

1.2.1　Linux 设备驱动程序的分类及特点

Linux 系统的设备分为字符设备(char device)、块设备(block device)和网络设备(network device)三类。字符设备是指可以直接读写、没有缓存的设备，如鼠标、键盘、串行口等。块设备指那些需要以块(如 512 字节)的方式读写的设备，如 IDE 硬盘、SCSI 硬盘、CD-ROM 等，读写都有缓存来支持，并且能够随机存取(random access)。网络设备是指 BSD socket 接口，如网卡等，它与字符设备和块设备有很大的不同，在 Linux 里进行了专门

的处理。Linux 的网络系统主要是基于 BSD UNIX 的 socket 机制，在系统和驱动程序之间定义有专门的数据结构(sk_buff)进行数据的传递，系统里支持对发送数据和接收数据的缓存，提供流量控制机制，提供对多协议的支持。

Linux 中的设备驱动程序有如下特点。

(1) 设备驱动程序是内核的一部分，如果驱动程序出错，则可能导致系统崩溃。

(2) 设备驱动程序必须为内核提供一个标准接口，这样，不论何种设备，内核都能以同样的方式进行访问。

(3) 设备驱动程序使用一些标准的内核服务，如内存分配等。

(4) 大多数的 Linux 设备驱动程序都既可以通过配置编译进内核，也可以只在需要时装载进内核、在不需要时从内核中卸载。

1.2.2 Linux 设备驱动程序与整个软件系统的关系

Linux 设备驱动程序与整个软件系统的关系如图 1-3 所示。除网络设备外，字符设备及块设备都被映射到 Linux 文件系统的文件和目录，通过文件系统的系统调用接口 open()、write()、read()、close()等函数即可访问字符设备和块设备。所有的字符设备和块设备都被统一地呈现给用户。块设备比字符设备复杂，在它上面会首先建立一个磁盘/Flash 文件系统，如 FAT、Ext3、YAFFS、JFFS 等。FAT、Ext3、YAFFS、JFFS 规范了文件和目录在存储介质上的组织。

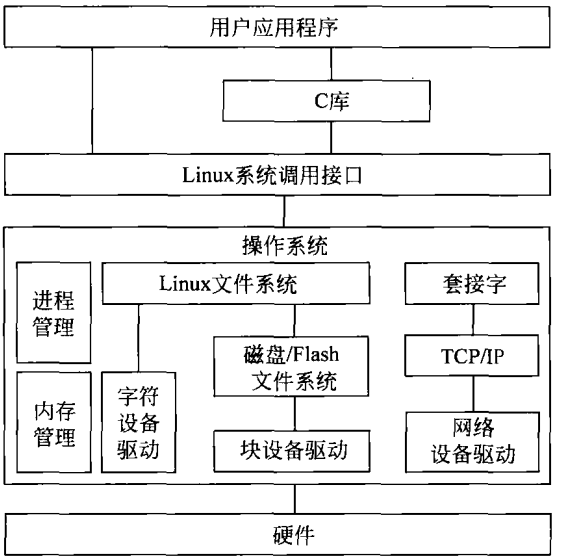

图 1-3 Linux 设备驱动与整个软硬件系统的关系

应用程序可以使用 Linux 的系统调用接口编程，也可以使用 C 库函数，出于代码可移植性的考虑，后者更值得推荐。C 库函数本身也通过系统调用接口而实现，如 C 库函数中的 fopen()、fwrite()、fread()、fclose()分别会调用操作系统 API 的 open()、write()、read()、close()函数。

1.3　Linux 设备驱动程序开发技术的学习方法

要学习 Linux 设备驱动程序的开发，读者首先应当具备一定的理论基础，包括：一定的硬件基础，如熟悉硬件的操作方法，熟悉程序查询、中断、DMA 三种 I/O 设备与主机信息传送的控制方式等；良好的 C 语言基础，能够灵活地利用 C 语言的结构体、指针、函数指针，熟悉内存的动态申请和释放；一定的 Linux 内核基础，了解设备驱动与内核的接口，尤其是对字符设备、块设备及网络设备等；良好的多任务并发控制和同步的基础，理解互斥、信号量、等待队列等并发与同步机制。

其次，实践是学习 Linux 驱动程序开发必不可少的环节，只有经常实践才能将书中的知识转换为实际的技能。本书包含了大量的实践内容，可遵循以下步骤来开展实验。

(1) 熟练使用 Linux 命令。由于大多数人首先习惯了 Windows 操作系统的图形界面，对 Linux 的命令产生畏惧或排斥心里，因此习惯性地使用 Linux 的图形界面进行各类操作。但是，只有熟练掌握 Linux 命令的使用方法才能提高实验的效率并进一步提高动手的兴趣。

(2) 熟悉一种编辑器的使用方法。在实验过程经常要用到编辑器编辑程序，因此熟悉一种编辑器的使用方法是必不可少的。

(3) 熟悉 Makefile 文件的编写方法。

(4) 上机练习不用访问实际硬件的模拟驱动程序的编译、加载、打开、读、写、关闭、卸载等过程，巩固对 Linux 驱动接口的理解。

(5) 在模拟驱动的基础上逐步增加对硬件操作的内容，进一步理解驱动程序各功能模块的编写方法。

只要具备一台安装了 Linux 2.4 内核的计算机，就能完成书中大部分实例，只有在第 6 章、第 8 章及第 10 章才需要 ARM 9 实验平台。当然，如果已经具有 ARM 9 实验平台，主计算机中可以安装其他版本的 Linux 系统。

习　题　1

1. 简述设备驱动程序的作用。
2. Linux 设备驱动程序有哪几类？
3. 联系自己，谈谈如何才能学好嵌入式 Linux 设备驱动程序开发技术。

第2章　Linux基本操作

嵌入式Linux开发环境的宿主机一般安装Linux操作系统，熟悉Linux的基本操作是开发Linux驱动程序的基础。针对Linux驱动程序开发的基本操作包括Linux常用命令的使用以及Linux程序的编辑和编译。本章首先讲述Linux常用命令的使用方法，其次讲述基于Linux的基本编程方法，为Linux驱动程序的编辑、编译做好准备。

2.1　Linux常用命令

2.1.1　文件及目录操作命令

1. pwd命令

该命令显示用户当前所处的目录。如果不知道自己当前所处的目录，就必须使用该命令。

用法：pwd

举例：

```
[root@localhost ~]# pwd
/root
```

说明当前目录是/root。

2. ls命令

该命令列出文件或子目录的信息。Linux系统用颜色来区分文件类别。默认时，蓝色代表目录，绿色代表可执行文件，红色代表压缩文件，浅蓝色代表链接文件，灰色代表其他文件。

用法：ls［参数］路径或文件名

参数选项

-a：显示所有的文件，包括以“.”开头的隐含文件。

-l：显示文件或子目录的详细信息，包括文件类型和权限、文件状态、文件所有者、文件所在组、文件大小、文件最近修改的时间、文件名等信息。

-i：显示每个文件的索引(节点)号。

举例：

(1) 列出当前目录中的文件，但不包含隐含文件。

```
[root@localhost /]# ls
arm2410s  dev   hello  lost+found  mnt   proc  selinux  tmp
bin       dir1  home   media       net   root  srv      usr
boot      etc   lib    misc        opt   sbin  sys      var
```