

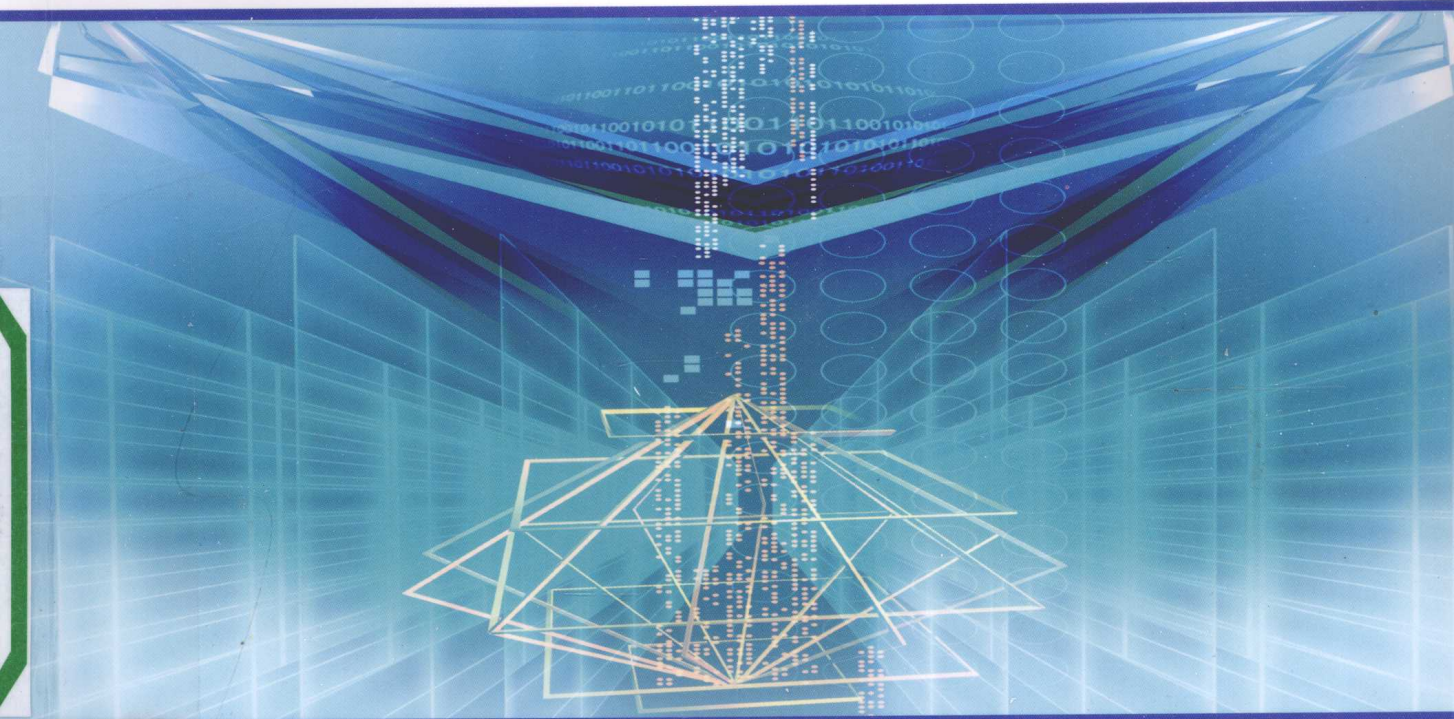
高等院校计算机教材系列

HZ BOOKS
华章教育

ARM嵌入式Linux系统 设计与开发

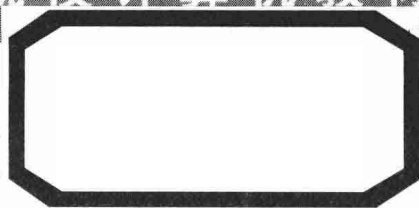
Embedded Linux System Design and
Development Based on ARM

俞辉 李永 刘凯 王晓虹 等编著



机械工业出版社
China Machine Press

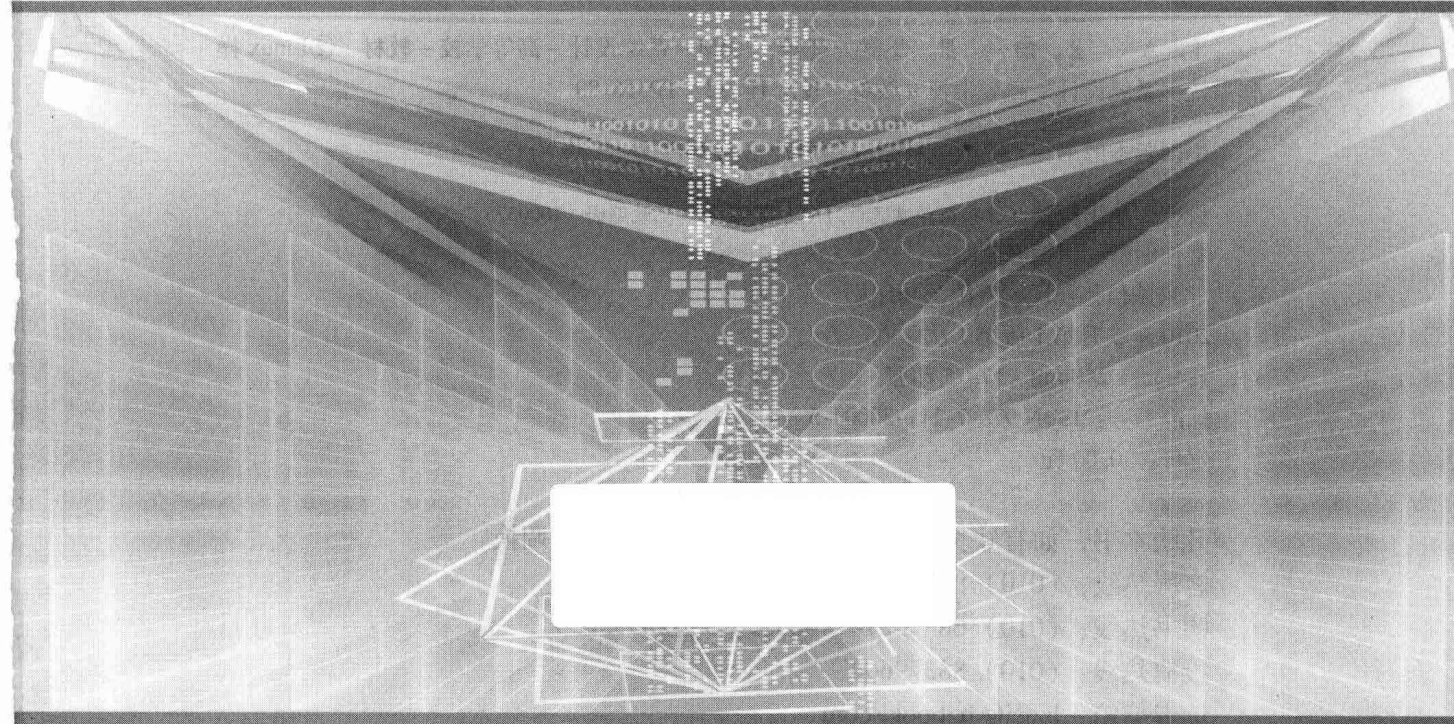
高等院校计算机教材系列



ARM嵌入式Linux系统 设计与开发

Embedded Linux System Design and
Development Based on ARM

俞辉 李永 刘凯 王晓虹 等编著



机械工业出版社
China Machine Press

本书全面介绍以 ARM-Linux 为操作系统的嵌入式系统设计与开发技术,秉承理论与实践相结合的指导思路,帮助读者快速跨入嵌入式系统开发的门槛。本书共 11 章,分三篇,其中,第一篇(第 1~3 章)主要介绍嵌入式硬件平台和软件系统的基础知识,第二篇(第 4~7 章)主要介绍基于 ARM-Linux 系统的驱动开发、软件移植、图形用户接口以及通信应用等内容,第三篇(第 8~11 章)主要介绍一些基于 ARM-Linux 的实际应用方案和案例。

本书内容丰富、概念清楚,适合作为高等院校计算机、电子、通信、软件工程等专业嵌入式课程的教材,也可供广大工程技术人员参考。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

ARM 嵌入式 Linux 系统设计与开发/俞辉等编著. —北京:机械工业出版社,2010.5
(高等院校计算机教材系列)

ISBN 978-7-111-30004-5

I. A… II. 俞… III. ①微处理器, ARM - 系统设计 - 高等学校 - 教材 ②Linux 操作系统 - 系统设计 - 高等学校 - 教材 IV. TP332 TP316.89

中国版本图书馆 CIP 数据核字(2010)第 037477 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:迟振春

北京诚信伟业印刷有限公司印刷

2010 年 5 月第 1 版第 1 次印刷

184mm×260mm·17.75 印张

标准书号:ISBN 978-7-111-30004-5

定价:30.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991; 88361066

购书热线:(010) 68326294; 88379649; 68995259

投稿热线:(010) 88379604

读者信箱:hzsj@hzbook.com

前 言

嵌入式 Linux 由于具有开源、网络功能强大、内核稳定高效等特性，在产品开发周期、产品的功能可扩充性、开发时的人力投入等方面都具有显著的优势，因此广泛应用于中低端智能电子设备中。而它与 ARM 的结合，更是一种主流的解决方案，嵌入式 Linux + ARM 已经广泛应用于机顶盒、掌上电脑、消费电子、MPC、网络设备、工业控制等领域，并且具有良好的市场前景。

本书全面介绍以 ARM-Linux 为操作系统的嵌入式系统设计与开发技术，秉承理论与实践相结合的指导思想，帮助读者快速跨入嵌入式系统开发的门槛。本书内容丰富、概念清楚、层次分明、通俗易懂，可以作为各类高等院校计算机、电子、通信、软件工程等专业高年级学生嵌入式 Linux 程序设计或嵌入式系统程序设计等课程的教材，也可作为相关工程技术人员和计算机爱好者学习嵌入式程序设计的参考书。

本书的编写目的与特色

本书在总结作者多年的教学经验的同时，为嵌入式 Linux 系统等相关课程提供配套的教材，通过实例，使读者能够较容易地掌握相关的嵌入式 Linux 程序设计方面的基本概念和技能；通过综合实例的学习和模拟，达到训练实际开发能力的目的。

“实践”和“创新”是嵌入式系统课程的两个主要特点。实践是整个嵌入式系统课程体系中重要的环节，而创新的思想贯穿整个嵌入式教学的始终，最后要通过案例的实现来对所学的嵌入式知识进行检验。

本书的主要内容与结构安排

本书共 11 章，分为三篇。第一篇（第 1~3 章）为“ARM 嵌入式 Linux 系统的基础与平台构建”，主要介绍嵌入式硬件平台和软件系统的基础知识，以期帮助读者达到构建嵌入式系统开发平台的目的。

第二篇（第 4~7 章）为“ARM 嵌入式 Linux 系统的驱动开发与软件应用”，主要介绍基于 ARM-Linux 系统的驱动开发、软件移植、图形用户接口以及通信应用等内容。这些内容构成了嵌入式软件系统相关实验的基础，涵盖了嵌入式 Linux 系统的几乎所有要点。在完成该篇内容的学习之后，即可进入 ARM-Linux 的实际应用部分。

第三篇（第 8~11 章）为“ARM 嵌入式 Linux 系统的解决方案与应用场合”，主要介绍一些基于 ARM-Linux 的实际应用方案和案例。本篇包含 OMAP5910 的相关应用、实时 Mobile Linux 系统、嵌入式 Linux 系统的设计与优化以及一个嵌入式 Linux 综合实例——数字油田生产监控系统。学生在了解该篇内容后，会对嵌入式系统目前的应用场合和解决方案有较深入的了解。

阅读本书时的注意事项

理论与实践相结合是本书的一大特色。若读者有相应的实验开发环境，建议动手验证书中安排的每一个环节，会取得良好的效果。具体实验可以参考《嵌入式 Linux 程序设计案例与实验教程》^①。读者需要特别注意：1) 某些代码需考虑平台的差异性，部分硬件相关代码需要修改；2) 相关软件的版本与依赖关系，同一软件（包括交叉编译器）的不同版本在编译环境、使用方式、编程要求上存在一些差异，具体要求与区别均需参考相关文档。

教学建议中的学时为教授各章全部知识点的所需学时，教师可以根据实际情况增删。

由于篇幅问题，我们将本书的一些代码和说明文档放到了网站上，读者可以登录 <http://www.hzbook.com> 下载。

本书在撰写过程中参考了许多相关公司的技术文档与网络资源，包括北京博创科技兴业有限公司。读者在阅读本书的同时，也可以查阅相关资料（参见本书最后的参考文献）。

结语

本书由俞辉执笔，李永、刘凯、王晓虹审定写作提纲。具体分工如下：中国石油大学（华东）李永编写了第 2、3、4、5 章，长沙民政学院刘凯编写了第 8 章，辽宁石油化工大学王晓虹编写了第 6 章，其余章节由俞辉编写，并负责统一校稿。另外，王爱国、黄晶等也参与了编写工作。

在本书的编写过程中，得到了许多同仁、北京博创科技兴业有限公司和机械工业出版社华章公司的大力支持和帮助，在此一并表示衷心的感谢。

由于时间仓促，加之水平有限，书中的不足之处在所难免，敬请读者批评指正。编者的电子邮箱是 bysj2004@126.com，欢迎来信交流。

编者

2010 年 1 月于中国石油大学（华东）青岛校区

① 该书已由机械工业出版社出版，书号为：978-7-111-26327-2。——编辑注

教学建议

章节	主要知识点	建议授课学时	实训目标与要求
第1章 嵌入式系统基础	<ol style="list-style-type: none"> 1. 嵌入式系统的概念与特点 2. 嵌入式系统开发基础 3. 嵌入式系统开发流程与方法 	上课：4 学时	<ol style="list-style-type: none"> 1. 了解有关嵌入式系统的概念、特点和开发基础 2. 了解关于嵌入式系统的开发流程和方法
第2章 ARM 与 Linux 基础	<ol style="list-style-type: none"> 1. ARM 体系结构介绍 2. XScale 体系结构介绍 3. Linux 系统的使用 4. vi 编辑器的使用 5. Shell 编程 6. Linux C 程序设计 7. 多线程库的使用 	<ol style="list-style-type: none"> 1. 上课：4 学时 2. 实验：8 学时 (根据需要, 可以删除部分实验) 	<ol style="list-style-type: none"> 1. 了解 ARM 体系结构相关知识 2. 掌握 Linux 相关命令的使用 3. 掌握 vi 使用和 Shell 编程 4. 熟练掌握 Linux C 程序开发, 具有使用多线程编程的能力
第3章 嵌入式 Linux 内核、 文件系统与存储	<ol style="list-style-type: none"> 1. 嵌入式引导系统 (Boot-Loader) 技术 2. Linux 内核定制、裁剪和添加 3. 文件系统的构建 	<ol style="list-style-type: none"> 1. 上课：4 学时 2. 实验：6 学时 	<ol style="list-style-type: none"> 1. 熟练掌握内核定制、裁剪和添加 2. 会使用 u-boot 的常用命令 3. 掌握 BusyBox 构建文件系统过程
第4章 嵌入式 Linux 接口 设计与驱动程序	<ol style="list-style-type: none"> 1. 驱动程序结构 2. AD 3. 触摸屏 4. 帧缓冲 5. V4L 	<ol style="list-style-type: none"> 1. 上课：6 学时 2. 实验：10 学时 (根据需要, 可以删除部分实验, 综合实验可以作为大作业或以小组的形式完成) 	<ol style="list-style-type: none"> 1. 熟练掌握驱动程序的结构 2. 读懂并理解 AD、触摸屏、帧缓冲、V4L 等驱动程序和测试程序的设计与编写
第5章 嵌入式 Linux 开源软件 移植与应用	<ol style="list-style-type: none"> 1. 嵌入式 Web 服务器 Go-Ahead 2. 嵌入式数据库 SQLite 3. 嵌入式媒体播放器 Mplayer 4. JVM phoneME 5. Web 服务 gSOAP 6. Web 浏览器 Konqueror 	<ol style="list-style-type: none"> 1. 上课：8 学时 2. 实验：16 学时 (根据需要, 可以删除部分实验和上课内容, 综合实验可以作为大作业或以小组的形式完成) 	<ol style="list-style-type: none"> 1. 掌握开源软件移植的一般过程 2. 理解相关软件的版本与依赖关系, 包括交叉编译器
第6章 嵌入式 Linux 图形 用户接口	<ol style="list-style-type: none"> 1. 嵌入式 GUI 介绍 2. Qt/E 介绍 3. Qt/E4 与键盘鼠标接口的映射 	<ol style="list-style-type: none"> 1. 上课：8 学时 2. 实验：8 学时 (根据需要, 可以删减部分内容, 综合实验可以以小组的形式完成, 软件可以扩充为最新版) 	<ol style="list-style-type: none"> 1. 理解 Qt/E4 移植的过程和相应的程序设计方法 2. 熟练掌握 Qt/E4 程序设计

(续)

章节	主要知识点	建议授课学时	实训目标与要求
第7章 嵌入式Linux下的 通信应用	1. 串口通信 2. 网络通信 3. 蓝牙通信 4. CAN总线通信 5. GPS和GPRS的应用	1. 上课: 8学时 2. 实验: 10学时 (根据需要, 可以删除部分实验和上课内容, 综合实验可以作为大作业或以小组的形式完成)	掌握几种常用的通信方式, 理解相关的程序设计思路与实现方式
第8章 OMAP5910与Linux Gateway	1. OMAP5910体系结构 2. Linux DSP Gateway 3. Gateway程序设计	1. 上课: 8学时 2. 实验: 8学时 (根据需要, 可以删除部分实验, 综合实验可以作为大作业或以小组的形式完成。若偏重DSP Gateway程序开发, 则此部分可以加强, 反之, 可以从略)	1. 理解OMAP5910体系结构与Linux DSP Gateway原理 2. 掌握Gateway程序设计与双核实际应用
第9章 实时Mobile Linux系统	1. LiMo相关介绍 2. Android相关介绍 3. Marvell PXA310介绍	1. 上课: 8学时 2. 实验: 8学时 (根据需要, 可以删除部分实验, 综合实验可以作为大作业或以小组的形式完成)	1. 了解LiMo、Android和Marvell等相关知识 2. 完成Android和Marvell相关实验
第10章 嵌入式Linux系统的 设计与优化	1. 嵌入式产品的生产链与关键因素 2. 嵌入式系统开发方法 3. 硬件选型对性能的影响 4. 软件设计对性能的影响 5. 软硬件协同设计方法	上课: 2学时	了解嵌入式系统设计和优化的相关知识
第11章 嵌入式Linux综合实例 ——数字油田生产监控系统的设计与实现	数字油田生产监控系统的设计与实现	此部分一般采用大作业或实习题目布置, 要求一小组学生在规定时间内完成相应系统, 并且撰写相应实习报告, 同时进行实际系统演示与PPT演讲。时间一般为4周左右	完成一个基于嵌入式系统的综合案例

说明: 上述每章学时安排为一般建议学时, 教师可以根据自己授课情况进行适当的调整。书中每章给出相关的习题, 教师可以根据学生掌握知识点的情况, 选择考察点作为习题。

目 录

前 言
教学建议

第一篇 ARM 嵌入式 Linux 系统的基础与平台构建

第 1 章 嵌入式系统基础	1
1.1 嵌入式系统的概念与特点	1
1.1.1 嵌入式系统的概念	1
1.1.2 嵌入式系统的特点	1
1.1.3 嵌入式系统的软硬件可裁剪	2
1.1.4 嵌入式系统与单片机、传统 PC 的区别	2
1.2 嵌入式系统开发基础	3
1.2.1 嵌入式系统的硬件基础	3
1.2.2 嵌入式系统的软件基础	10
1.3 嵌入式系统开发流程与方法	14
1.3.1 嵌入式系统的抽象体系 结构	14
1.3.2 实时软件分析设计方法 DARTS	15
1.3.3 嵌入式系统的开发流程	16
1.3.4 嵌入式 Linux 系统的程序 部署	17
习题	18
第 2 章 ARM 与 Linux 基础	19
2.1 ARM 概述	19
2.1.1 ARM 体系结构	19
2.1.2 XScale 体系结构	24
2.2 Linux 系统的使用	27
2.2.1 Linux 的使用基础	27
2.2.2 Linux 内核与结构	37
2.2.3 嵌入式 Linux 的启动过程	41

2.3 Linux C 程序设计	42
2.3.1 Linux C 程序设计基础	43
2.3.2 调试工具	45
2.3.3 Linux 多线程库编程	52
习题	56
第 3 章 嵌入式 Linux 内核、文件系统 与存储	57
3.1 嵌入式引导系统技术	57
3.1.1 概述	57
3.1.2 u-boot 目录结构	58
3.1.3 u-boot 启动流程及相关 代码分析	59
3.1.4 u-boot 的常用命令	67
3.2 Linux 内核定制、裁剪和添加	68
3.2.1 概述	68
3.2.2 配置文件和配置工具	69
3.2.3 内核的编译命令	71
3.2.4 内核的烧写过程	71
3.3 文件系统的构建	73
3.3.1 概述	73
3.3.2 BusyBox 的移植	74
3.4 关于 2.6 版本内核的几点改进	78
习题	79

第二篇 ARM 嵌入式 Linux 系统的驱动开发与软件应用

第 4 章 嵌入式 Linux 接口设计与 驱动程序	81
4.1 驱动程序设计基础	81
4.1.1 Linux 驱动程序简介	81
4.1.2 开发驱动程序的方法	81
4.1.3 设备驱动程序的分类	82
4.1.4 主设备号和次设备号	82

4.1.5	udev 介绍	83	5.3.1	gSOAP 介绍	134
4.2	虚拟驱动模块实验	90	5.3.2	gSOAP 裁剪	137
4.2.1	内核的配置	90	5.3.3	gSOAP 应用	140
4.2.2	编写驱动程序	94	5.3.4	gSOAP 移植	145
4.2.3	测试程序和 Makefile	96	5.4	嵌入式数据库 SQLite 的移植 与应用	147
4.2.4	测试程序的执行过程	96	5.4.1	SQLite 介绍	147
4.3	AD 接口驱动程序	97	5.4.2	SQLite 在 ARM 平台上 的移植	148
4.3.1	AD 转换器	97	5.4.3	SQLite 的使用	151
4.3.2	AD 转换有关参数	99	5.5	播放器 Mplayer 的移植	154
4.3.3	ARM 自带的 AD 转换装置	99	5.5.1	Mplayer 介绍	154
4.3.4	程序说明	100	5.5.2	Mplayer 在 ARM 平台上 的移植	154
4.4	触摸屏接口设计与驱动	104	5.6	J2ME-phoneME 的移植	157
4.4.1	触摸屏的工作原理	104	5.6.1	phoneME 介绍	157
4.4.2	S3C2410 芯片的触摸屏 相关配置寄存器	105	5.6.2	软件移植过程	157
4.4.3	程序说明	109	5.7	嵌入式浏览器 Konqueror 的移植	158
4.5	帧缓冲与 LCD 驱动移植	110	5.7.1	Konqueror 介绍	158
4.5.1	帧缓冲机制介绍	110	5.7.2	软件移植过程	159
4.5.2	LCD 简介	111	习题		161
4.5.3	LCD 控制器初始化时所需 的数据结构	111	第 6 章	嵌入式 Linux 图形用户接口	162
4.5.4	移植 LCD 驱动程序	115	6.1	嵌入式 GUI 简介	162
4.6	V4L 程序设计	119	6.1.1	嵌入式 GUI 的特点	162
4.6.1	V4L 概述	119	6.1.2	嵌入式 GUI 的种类	162
4.6.2	视频驱动实验	121	6.2	嵌入式 GUI——Qt	163
习题		128	6.2.1	Qt 与 Qt/Embedded 简介	163
第 5 章	嵌入式 Linux 开源软件移植 与应用	129	6.2.2	Qt 的特点	163
5.1	开源软件移植的一般过程	129	6.2.3	Qt 的执行过程	164
5.1.1	软件移植的概念	129	6.2.4	Qt 的插槽机制	166
5.1.2	软件移植过程	129	6.2.5	一个完整的 Qt 程序	166
5.2	嵌入式 WebServer GoAhead 的 移植与应用	130	6.2.6	Qt Designer 介绍	166
5.2.1	嵌入式 Web 服务器	130	6.3	Qt/E4 与键盘鼠标接口的映射	167
5.2.2	GoAhead 介绍	131	6.3.1	Qt/E4 的移植过程	167
5.2.3	GoAhead 在 ARM 平台上 的移植	131	6.3.2	Qt/E4 键盘鼠标接口的 工作原理	170
5.2.4	页面操作	133	6.3.3	Qt/E4 与 Linux PS/2 键盘鼠标驱动的衔接	172
5.3	嵌入式 WebService gSOAP 的 移植与应用	134			

6.3.4 Qt/E4 键盘驱动映射的实现	172	8.1.2 DSP 子系统	208
习题	174	8.2 Linux DSP Gateway	210
第7章 嵌入式 Linux 下的通信应用	175	8.2.1 DSP Gateway 的由来	210
7.1 嵌入式 Linux 下的串口通信	175	8.2.2 DSP Gateway 的 Mailbox 机制	210
7.1.1 串口简介	175	8.2.3 通信缓冲	210
7.1.2 串口编程	175	8.2.4 Mailbox 命令协议	213
7.1.3 串口编程应用实例	177	8.2.5 DSP Gateway 的设备接口	214
7.2 嵌入式 Linux 网络编程	179	8.3 OMAP5910 图像处理	215
7.2.1 网络通信	179	8.3.1 图片格式	215
7.2.2 Socket 简介	179	8.3.2 数字图像算法	216
7.2.3 网络编程	180	8.4 基于 OMAP 的加密终端的实现 (软件部分)	218
7.2.4 网络编程应用实例	182	8.4.1 加密算法简介	218
7.3 嵌入式蓝牙技术	185	8.4.2 加密终端的具体实现	220
7.3.1 蓝牙技术	185	习题	221
7.3.2 蓝牙体系结构	185	第9章 实时 Mobile Linux 系统	222
7.3.3 蓝牙通信网络	186	9.1 LiMo 基金会	222
7.3.4 Linux Bluetooth 软件层	186	9.1.1 LiMo 基金会介绍	222
7.3.5 USB 适配器	186	9.1.2 LiMo 手机解决方案的软件体系结构	222
7.3.6 蓝牙应用实例	187	9.2 开放手机联盟	224
7.4 CAN 总线	191	9.2.1 开放手机联盟与 Android	224
7.4.1 CAN 总线简介	191	9.2.2 Android 应用程序结构分析	227
7.4.2 CAN 总线硬件特征	191	9.2.3 Android 开发环境搭建	229
7.4.3 CAN 控制器驱动	194	9.2.4 Android 应用程序开发——Hello World	234
7.4.4 CAN 总线编程实例	196	9.3 Mobile Linux 系统的构建实例	236
7.5 GPS 和 GPRS 的应用	199	9.3.1 Marvell PXA310 简介	236
7.5.1 GPS 原理与应用	199	9.3.2 基于 PXA310 的 Mobile Linux 系统构建过程	237
7.5.2 GPS 编程实例	200	习题	240
7.5.3 GPRS 原理与应用	203	第10章 嵌入式 Linux 系统的设计与优化	241
7.5.4 GPRS 编程实例	204	10.1 嵌入式产品的生产链与关键因素	241
习题	205	10.2 嵌入式系统开发方法	241
		10.2.1 软硬件“分离”设计方法	241
第三篇 ARM 嵌入式 Linux 系统的解决方案与应用场合			
第8章 OMAP5910 与 Linux Gateway	207		
8.1 OMAP5910 体系结构	207		
8.1.1 MPU 子系统	207		

10.2.2	非形式化建模方法与形式化建模方法	242	10.5.2	嵌入式软硬件协同设计方法	247
10.2.3	结构化设计范式与面向对象设计范式	243	习题		249
10.3	硬件选型对性能的影响	243	第 11 章 嵌入式 Linux 综合实例		
10.3.1	硬件平台的分类	244	——数字油田生产监控系统的设计与实现		250
10.3.2	嵌入式处理器的选择	244	11.1 系统概述		250
10.3.3	外围设备的选择	244	11.1.1 系统结构和功能简介		250
10.3.4	硬件成本	245	11.1.2 软硬件环境		252
10.4	软件设计对性能的影响	245	11.2 系统设计		252
10.4.1	嵌入式软件开发的特点	245	11.2.1 系统总体方案设计		252
10.4.2	嵌入式软件开发中存在的问题	245	11.2.2 无线传感器节点设计		253
10.4.3	常用嵌入式软件设计方法	246	11.2.3 远程测控终端 RTU 设计		254
10.5	软硬件协同设计方法	247	11.3 软件设计		259
10.5.1	传统设计方法的局限性	247	11.3.1 远程测控终端 RTU 软件设计		259
			11.3.2 监控中心软件设计		266
			参考文献		271

第一篇 ARM 嵌入式 Linux 系统 的基础与平台构建

第 1 章 嵌入式系统基础

本章主要阐述嵌入式系统的基础知识，包括嵌入式系统的概念与特点、嵌入式系统的开发基础以及嵌入式系统的开发流程与方法。

1.1 嵌入式系统的概念与特点

1.1.1 嵌入式系统的概念

根据 IEEE（国际电气与电子工程师协会）的定义，嵌入式系统是“devices used to control, monitor, or assist the operation of equipment, machinery or plants”（即“控制、监视或者辅助机器和设备运行的装置”之意）。这主要是从应用上加以定义的，表达嵌入式系统是一个软硬件综合体的事实。

而随着嵌入式技术的发展，嵌入式系统的定义已远远超出了这个范畴，小到一块带有脉搏、血压监测的智能运动型手表，大到外星球上使用的智能探测设备，都是由嵌入式系统构成的。业界有多种不同的关于嵌入式系统（Embedded System）的定义，被大多数人所接受的是根据嵌入式系统的特点下的定义：“以应用为中心、以计算机技术为基础、软硬件可裁剪，功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统”。

该定义强调软硬件可裁剪、专用计算机系统的特点，这也是嵌入式系统与通用计算机平台最为显著的差别。嵌入式系统在工业生产控制、智能仪表、信息家电、网络通信等领域都有着广泛的应用。它是应用于特定环境下、针对特定用途设计的系统。它的硬件和软件都必须高效率地设计，力争在较少的资源上实现更高的性能。

1.1.2 嵌入式系统的特点

与通用计算机相比，嵌入式系统具有以下显著特点：

- 嵌入式系统是专用的计算机系统，因此通常含有面向某种特定应用的嵌入式 CPU。
- 嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合的产物，因此可以说嵌入式系统不隶属于任何一门学科，但是却横跨了多门学科。
- 嵌入式系统的硬件和软件设计必须达到功耗低、体积小、集成度高、成本低的目标。
- 嵌入式系统和具体应用有机结合在一起，运行环境差异大，且具有较长的生命周期。

- 为了提高执行速度和系统可靠性，嵌入式软件系统一般都固化在非易失性存储器中。
- 嵌入式系统本身不具有自举开发能力，因此需要进行交叉编译，即必须有一套开发工具和环境才能进行开发。

1.1.3 嵌入式系统的软硬件可裁剪

前面已经提到，嵌入式系统是软硬件可裁剪的专用计算机系统。而软硬件可裁剪也是嵌入式系统最具代表性的特征之一。开发人员可以针对不同用户、不同应用定制嵌入式系统，使得嵌入式系统比通用计算机系统更灵活、更简单、更可靠。与这些优点相对应的就是嵌入式系统对开发人员的要求比较高，开发人员必须深入理解硬件接口技术和操作系统技术。

1. 硬件裁剪

由于嵌入式系统是面向用户、面向应用、面向产品的，因此在开发过程中必须考虑成本、体积、功耗、可靠性、实时性、电磁兼容性等各方面的制约。否则，即使发布了产品也不一定会有很好的市场。低成本是所有商家追求的目标，它在嵌入式系统中的表现尤其突出。例如，对 MP3 来说，串口、网口等接口不是必需的，因此在设计硬件电路时就可以裁剪掉。而在工业控制领域，考虑到对嵌入式系统的可靠性和实时性的要求，在设计硬件电路时往往要裁剪掉影响系统可靠性的电路，或者使用特殊的措施对这部分电路进行保护，因此又有可能增加一定的成本。所以，裁剪是相对而言，要针对用户的需求，在成本、体积、功耗等多方面进行权衡，这样才能使嵌入式产品更具市场竞争力。

2. 软件裁剪

软件裁剪也是嵌入式系统中常见的技术，其中最具代表性的当属操作系统裁剪。目前，嵌入式操作系统的种类很多，比较常见的有嵌入式 Linux、Windows CE、VxWorks、uClinux、uC/OS-II 等。由于操作系统并不是针对某个应用产品而开发的，且嵌入式设备上的存储资源一般比较有限，因此在将操作系统应用到某个嵌入式产品的过程中，需要进行裁剪，使之更适合特定的需求。

在定制 Linux 内核时，可以在编译内核之前通过 `make menuconfig` 命令对内核进行裁剪，之后编译生成压缩的映像。而对于 Windows CE 操作系统来说，可以利用 Platform Builder 开发环境对内核进行定制。风河公司也推出了 workbench 开发工具，使开发人员可以方便地进行基于 VxWorks 操作系统的定制与开发。虽然有很多工具使操作系统的裁剪工作变得更加方便，但是对于开发人员来说，仅仅会使用这些工具还远远不够。开发人员必须对操作系统的各个功能模块有比较全面的了解，只有这样才能定制出一款精简且优秀的操作系统。

除了系统软件，应用软件也需要进行相关裁剪。目前应用软件开发越来越模块化，尤其是一些比较通用的应用软件。在将这些软件移植到特定的嵌入式系统中时，有时并不需要所有的软件功能，因此需要进行一定的裁剪。软件移植裁剪的具体内容，将在本书第 5 章进行详细介绍。

1.1.4 嵌入式系统与单片机、传统 PC 的区别

嵌入式系统是嵌入式领域中较高端的产物，其硬件基于一个以高性能处理器为中心的核心板开发专有功能，而软件则基于一个多任务操作系统进行高级编程。换句话说，这里所指的嵌入式系统和嵌入式低端产品最显著的区别就是其操作系统的存在。借由操作系统，该综合平台的处理能力是以往的嵌入式单片机所根本无法比拟的。本书的“嵌入式系统”特指这种包括操作系统

的综合嵌入式平台。

一款嵌入式系统成品可以理解为一款专用计算机系统，因此相对于传统 PC 来说，嵌入式系统由于其专用型也存在着很大的区别。嵌入式系统和传统 PC 的区别如表 1-1 所示。

表 1-1 嵌入式系统和传统 PC 的区别

	嵌入式系统	PC
引导代码	BootLoader 引导，针对不同电路板进行移植	主板的 BIOS 进行引导，因此无需改动
操作系统	Windows CE、VxWorks、Linux 等，需要移植	Windows、Linux 等，不需要移植
驱动程序	每个设备驱动必须重新开发和移植，一般不能直接下载使用	操作系统含有大多数驱动程序，从网上下载直接使用
协议栈	需要移植	操作系统包括，或第三方提供
开发环境	借助服务器进行交叉编译	在本机就可开发调试
仿真器	需要	不需要

1.2 嵌入式系统开发基础

一个完整的嵌入式系统包括硬件部分和软件部分，这两个部分的设计开发都要遵循各自的特点。下面将分别介绍嵌入式系统的硬件基础和软件基础。

1.2.1 嵌入式系统的硬件基础

1. 嵌入式系统的硬件组成要素

硬件是嵌入式系统底层基础的部分，对于不同的嵌入式系统，其硬件的实现基本不同。但是，一般的嵌入式系统硬件可分为如下几个部分：嵌入式微处理器、存储器、输入/输出设备、通信与扩展接口。

(1) 嵌入式微处理器

嵌入式微处理器是嵌入式系统的核心部件，是决定嵌入式系统功能的主要因素，也决定了嵌入式系统的应用范围和开发复杂度。嵌入式微处理器一般包括单片机、嵌入式 DSP 处理器、嵌入式片上系统和嵌入式微处理器这几种。其中，单片机（Microcontroller Unit）由于单片化，体积非常小，因此功耗和成本下降、可靠性大大提高。单片机在嵌入式设备中的应用最为广泛，是目前嵌入式系统工业的主流。DSP 处理器专门用于信号处理，其在系统结构和指令算法方面进行了特殊设计，在数字滤波、FFT、谱分析等各种仪器上都获得了大规模的应用。嵌入式片上系统（System on Chip, SoC）就是一种电路系统，它包含诸多功能单元，并且结合了许多功能区块，将这些功能做在一个芯片上。而嵌入式微处理器（Microprocessor Unit, MPU）由通用计算机中的 CPU 演变而来。MPU 与计算机处理器的不同之处在于，在实际的嵌入式应用中，只保留和嵌入式应用紧密相关的功能硬件，而去除其他的冗余功能部分，这样就以最低的功耗和资源满足了嵌入式应用的特殊要求。MPU 的最大特点就是功能相当强大，由于它保留了通用计算机的 CPU 特征，因此可以运行操作系统。MPU 目前主要有 X86、Am186/88、ARM、MIPS、COLD-FIRE、PowerPC68K 等系列，其中以 ARM 芯片最为著名。

嵌入式微处理器与通用处理器的最大区别在于，嵌入式微处理器一般工作于特定的系统中，通常把通用处理器中许多由板卡完成的任务集成在芯片内部，从而有利于实现嵌入式系统设计小型化、高效率、高可靠性等特点。有时，为了满足嵌入式应用的特殊要求，嵌入式微处理器还会在工作温度、抗电磁干扰、可靠性等方面加以增强。一般来说，嵌入式微处理器具有以下特点：

- 具有较强的实时多任务处理能力，能完成多任务并且中断响应时间较短，从而使内部的代码和实时内核的执行时间降到最低。
- 具有强大的存储区保护功能。这是由于嵌入式系统的软件结构已实现模块化，为了避免在软件模块之间出现错误操作和不必要的交叉作用，需要设计强大的存储区保护功能，同时也有利于软件诊断。
- 可扩展的处理器结构。一般在处理器内部都留有很多扩展接口，以便对应用进行扩展。
- 提供丰富的调试功能。嵌入式系统的开发过程大部分使用交叉编译，丰富的调试接口可以使嵌入式系统的开发更加方便。
- 嵌入式微处理器必须功耗很低，对于那些靠电池供电的便携式系统更是如此，有的系统要求功耗达到 mW 甚至 μ W 级。许多嵌入式微处理器提供几种工作模式，如正常工作模式、备用模式、省电模式等，从而为嵌入式系统提供了灵活性，可以满足嵌入式系统对低功耗的要求。

(2) 存储器

存储器是嵌入式系统的“记忆”部件，是嵌入式系统的另一个重要组成部分。在嵌入式系统中，按照信息的易失性，把存储器分为易失性存储器和非易失性存储器。易失性存储器是指存储器中的信息掉电即消失，不能持久保存；相比之下，非易失性存储器中的数据则不会在掉电后消失，可以持久保存。下面进行详细介绍。

1) 易失性存储器：在嵌入式系统中，最常用的易失性存储器是随机存储器（Random Access Memory, RAM）。按照存储机理的不同，RAM 又可分为动态 RAM（Dynamic RAM, DRAM）和静态 RAM（Static RAM, SRAM）。

DRAM 中的每个存储单元都有一个晶体管和一个电容，数据存储电容中。电容会由于漏电而导致电荷丢失，因此电容存储的数据（电荷）不能长久保存。为了将数据保存在存储器中，必须定期给电容补充电荷，以避免存储数据的丢失。这种操作称为再生或刷新（refresh）。DRAM 要不断地刷新，才能使保存的数据不会丢失。而在刷新过程中 DRAM 是停止一切操作的，这使得 DRAM 与微处理器的配合大打折扣，影响了高速微处理器的使用效率。但由于 DRAM 所用的元件主要是一个 CMOS 电路和一个电容，因此具有设计简单、集成度高、容量大、价格低的特点。目前，主要采用 DRAM 作为计算机系统的内存。

SRAM 不像 DRAM 那样需要不断地刷新，只要电源持续供电，其上的信息就不会消失。也正是因为 SRAM 不用刷新，所以其存取速度很快。但由于它的存储单元是由 6 个 MOS 管组成的，因此功耗比较大，而且集成度受限制，容量不容易做得很大，目前主要用作高速缓存（Cache）。在主板上或微处理器内部加上 Cache，微处理器就会把一些需要重复且大量存取的数据放在 Cache 中，这样就可以提高微处理器的工作效率，缓解处理器和 DRAM 之间在数据存取速度上的矛盾。

2) 非易失性存储器：在嵌入式系统中，常用的非易失性存储器有只读存储器（ROM）、Flash 和微硬盘（MicroDrive）。

在嵌入式系统中使用的只读存储器一般不是普通的只读存储器，而是可编程只读存储器（Programmable ROM, PROM）或可擦写可编程只读存储器（Erasable Programmable ROM, EPROM），里面主要存储该嵌入式产品所要使用的一些系统控制程序，嵌入式微处理器根据这些控制程序执行指令所指定的动作。

随着用户需求的提高和新应用的出现，嵌入式系统需要处理的数据越来越多，相应地需要的

存储空间也越来越多。普通存储器庞大的体积和可观的能耗，对嵌入式系统来说是不可接受的。在这种情况下，Flash 或微硬盘以可靠性高、体积小、重量轻、容量较大等特点而广泛应用于嵌入式系统中。

Flash 存储器是在 EPROM 和 EEPROM 基础上发展起来的非易失性存储器。它基于电荷存储原理来存储数据，其最大的特性就是能在切断供电电源后保持所存储的数据，且时间长达 10 年之久。Flash 存储器能在不离开电路板或所在设备的情况下，实施擦除和再编程操作。

Flash 存储器按其内部存储矩阵结构的不同可分为 NOR 型和 NAND 型两种。NOR 型 Flash 的存储矩阵采用“或非”结构的栅格实现，而 NAND 型 Flash 的存储矩阵采用“与非”结构的栅格实现。因此，NOR 型 Flash 与 NAND 型 Flash 在存储容量、数据的读/写方式以及相应的性能等方面有极大的不同。表 1-2 为两种 Flash 存储器的比较。

表 1-2 NOR 型 Flash 与 NAND 型 Flash 的比较

项目	NOR	NAND
存储容量	1 ~ 128Mb	8Mb ~ 2Gb
擦除块大小	64 ~ 256Kb	16 ~ 128Kb, 每块被分为若干个 512 字节的页面, 每页面含有 15 字节附加存储区
写入方式	按字节随机写入	以页面为单位写入
读取方式	按字节随机读取	按字节随机读取或按页面读取
写入速度	慢	快
读取速度	快	慢
擦除时间	2 ~ 5 s/块	2 ~ 5 ms/块
坏块比例	低	高, 需要 EDC/ECC 算法
辅助软件	无	需要程序驱动

虽然 Flash 存储器的容量越来越大，存储速率也有所提升，但是面对很多大容量的应用时，还是显得有些力不从心。这时微硬盘设备的优势就体现出来了。多大尺寸的硬盘才算微硬盘，现在并没有一个比较权威的规定，一般把 2.5 英寸以下的硬盘称为微硬盘。目前，微硬盘主要有三种尺寸：1.8 英寸、1 英寸和 0.85 英寸。微硬盘技术很复杂，涉及磁性材料、精密机械、微动力马达、光电材料、芯片封装、电子模组加工等专业技术，并不是简单地把普通硬盘小型化。目前主流微硬盘的容量是 1 ~ 4GB，更大容量的产品（如 15GB 和 60GB）也已出现。主流微硬盘的转速可达 3600 ~ 4200 rpm（转/分），而且加入了 128KB 容量的数据缓存，所以读/写速度很快。

与 Flash 存储器相比，微硬盘的最大优势是容量大而价格相对较低；缺点是稳定性、抗震性不如 Flash 存储器，发热量、耗电量较大，容易受磁和静电影响。微硬盘技术还需要不断地完善，才能在嵌入式系统中得到更广泛的应用。表 1-3 对 Flash 和微硬盘的特点进行了比较。

表 1-3 Flash 和微硬盘的比较

项目	Flash	微硬盘
存储容量	NOR Flash 为 1 ~ 128Mb, NAND Flash 为 8Mb ~ 2Gb	1 ~ 4GB 或更大
读取速度	较快	快
写入速度	较快	快
擦除时间	较慢	不需要擦除
体积	很小	较小
重量	轻	较轻
功耗	很低	较大

(续)

项目	Flash	微硬盘
单位成本	低	很低
可靠性	高	易受磁和静电影响
抗震性	好	较差

(3) 输入/输出设备

1) 输入设备: 计算机要按人的要求进行工作, 就必须能够接收人的命令, 将完成各种工作所需要的原始数据送入计算机系统内。承担这些任务的就是计算机的输入设备。输入设备的功能是输入程序、数据、指令及各种字符信息。操作者可通过输入设备对系统发出指令, 与系统“对话”, 输入设备是人与系统之间进行信息交换的主要装置之一。嵌入式系统同样也离不开输入设备。通用计算机系统的输入设备种类不是很多, 比较常用的有键盘、鼠标、触摸屏和扫描仪等。相比之下, 嵌入式系统的输入设备就是五花八门了, 比较常见的有按键、键盘、鼠标、手柄、触摸屏、声控/光控开关等。按照输入设备实现机理的不同, 嵌入式系统的输入设备可分为机械式、触控式、声光式三类。

2) 输出设备: 输出设备是计算机系统向用户传送计算、处理信息结果的部件。输出设备的功能是输出处理结果, 把系统对信息进行加工、处理的结果显示或打印出来。通用计算机系统中常见的输出设备有显示器、打印机、绘图仪等, 在嵌入式系统中除了以上几种之外, 还有 LED 指示灯以及扬声器等音频设备。

显示器是最常用的输出设备, 可分为球面显示器和平板显示器两类。球面显示器中最常见的就是传统的 CRT 显示器。由于球面显示器的体积都比较大, 因此不适用于嵌入式系统的显示。在嵌入式系统中通常使用平板显示器来输出、显示数据。在平板显示技术中最重要的就是液晶显示技术。现在, 液晶显示器 (Liquid Crystal Display, LCD) 已广泛用于嵌入式系统。

(4) 通信与扩展接口

嵌入式系统与外设及其他计算机系统通信时, 若要进行功能扩展, 就离不开通信与扩展接口。接口是在主板和某一类外设之间的适配电路, 可以解决主板和外设之间在工作速度、数据格式和电压等级上的相互匹配问题。接口在嵌入式系统中扮演着沟通外部世界的“桥梁”作用。嵌入式系统中接口的形式和方法多种多样。

嵌入式系统的通信接口可分为有线接口和无线接口两种。有线接口在设计上必须考虑以下几个因素: 首先是电位匹配, 就是嵌入式系统微处理器所提供的接收信号的电位必须与相连接的外围装置相同, 否则一方电位较高, 会导致电路的毁损, 或者是电位不同, 无法准确判断所代表的正确信号。其次就是要防止连接时干扰的产生。当接口连接时, 若是连接不完全, 会出现信号传递受阻的问题。最后要考虑驱动电位的问题, 有的外设需要比较大的功率来驱动, 若功率太低, 则外设无法正常操作。在嵌入式系统中比较常用的有线接口有 RS-232 接口、USB 接口、IEEE-1394 接口及 RJ-45 接口等。

近年来, 无线接口在嵌入式系统中的应用越来越广泛, 最常见的无线接口就是红外线接口。在手机或个人数字助理等嵌入式产品中经常会看到一个偏黑透明的小方格, 这个小方格就是红外线接口, 用户可通过红外线把嵌入式系统内的数据传输到外设上, 或把数据从外设发送到嵌入式系统内。随着蓝牙技术的逐渐成熟, 蓝牙接口也在嵌入式系统中得到了应用。