HZ BOOKS
华章科技

Linux/Unix
技术丛书

深度实践嵌入式Linux系统移植

范展源 刘韬 著

Inside Explore the
Embedded Linux Systems Porting

- 嵌入式Linux领域的里程碑之作，由有多年实践经验的资深嵌入式Linux专家撰写，深度与实践性兼备
- 从源码实现和工程实践两个维度深度讲解了u-boot、Linux内核、驱动和应用程序的移植原理和方法

机械工业出版社
China Machine Press

深度实践嵌入式Linux系统移植

Inside Explore the
Embedded Linux Systems Porting

范展源 刘韬 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

深度实践嵌入式 Linux 系统移植 / 范展源，刘韬著 . —北京：机械工业出版社，2015.5
（Linux/Unix 技术丛书）

ISBN 978-7-111-49791-2

I. 深… II. ① 范… ② 刘… III. Linux 操作系统 IV. TP316.89

中国版本图书馆 CIP 数据核字（2015）第 061167 号

深度实践嵌入式 Linux 系统移植

出版发行：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：姜 影　　责任校对：董纪丽

印　刷：三河市宏图印务有限公司　　版　次：2015 年 5 月第 1 版第 1 次印刷

开　本：186mm × 240mm 1/16　　印　张：51.5

书　号：ISBN 978-7-111-49791-2　　定　价：129.00 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　　投稿热线：（010）88379604

购书热线：（010）68326294 88379649 68995259　　读者信箱：hzjsj@hzbook.com

Preface 前　言

为什么要写这本书

近年来，以嵌入式技术为基础的物联网产业正如火如荼地发展，就业需求和薪资待遇年年看涨。许多学校和培训机构纷纷开设了嵌入式相关的课程，但这些课程的实践环节往往存在以下几个方面的问题：

- 过于偏重理论，缺乏对实践环节的指导。
- 有实践环节，但内容过于简单，流于表面。
- 有实践环节，但移植实践部分讨论的内容不完整。

由于存在上述的问题，通过这些课程培养出来的学生往往满足不了企业的真正需求。因此，相关学员也很难圆高薪就业的梦想。

本书力求为上述三个问题寻求解决方案：第一，在注重理论知识的前提下，坚持以实践环节为本书内容的主体，从实践中提炼理论知识，以源码分析作为移植实践的依据；第二，绝不做浅尝辄止的实践，绝不停留于只求在某个开发板上做出某个简单的效果，而是从对多个开发板的移植实践中提炼出更一般性的方法；第三，力求做到功能完善的系统移植。虽然嵌入式系统往往需要根据需求对功能进行裁剪，但为了打造一本面向广大嵌入式工程师和爱好者的书，笔者还是决定向读者介绍更多功能的移植方法，以让本书可以作为一本工具书常备、常用。

读者对象

这里可以根据软件需求来划分本书的读者：

- Linux 系统爱好者
- 嵌入式 u-boot 移植工程师
- 嵌入式 Linux 移植工程师
- Linux 驱动开发工程师
- 嵌入式应用开发工程师
- 高校相关课程的教师与学生

- 参加嵌入式职业培训的人员

如何阅读本书

本书共 23 章，分为四篇。

第一篇为绪论篇（第 1 章）。本篇只有一章内容，从整体上简单地介绍了嵌入式系统架构，并探讨了系统移植环境的搭建方法。本章中的所有内容均以介绍和指导实践为主，读者仅需了解相关的概念，之后自会在移植实践过程中有深刻体会。

第二篇为 u-boot 移植篇（第 2 ～ 5 章）。从绪论篇中得知，嵌入式系统的移植首先需要完成的工作是启动加载程序的移植，由于本书是以实践为目的，因此选择了嵌入式系统最为常用的 u-boot 作为移植的依据。本篇首先介绍 u-boot 工程与编译系统以及 u-boot 启动流程等基础知识，并深入源码进行详细分析。读者在掌握这些基础知识后，可以参照使用 ARM9/S3C2440 芯片的开发板和使用 ARM11/S3C6410 芯片的开发板这两个实例进行系统的 u-boot 移植实战。由于力求深入与完整，移植实战涵盖了从最初的启动代码到各种常见驱动的移植，并在最后整合所有的移植功能，完成一个功能强大的启动菜单。所有的移植代码和相关资源均可在本书配套资源文件中找到。

第三篇为 Linux 内核移植篇（第 6 ～ 18 章）。本篇是本书的核心内容，与第二篇类似，在正式开始内核移植前，首先介绍 Linux 内核工程及其编译系统、Linux 内核相关的调试技术，以及 Linux 内核的启动流程，为内核移植的讲解打下基础。然后以使用 ARM9/S3C2440 芯片的开发板和使用 ARM11/S3C6410 芯片的开发板为例，进行彻底的系统移植。

第四篇为应用程序移植篇（第 19 ～ 23 章）。在完成操作系统底层移植后，为了能在开发板上实现具体的应用，安排了本篇内容。本篇将从嵌入式图形界面移植、嵌入式多媒体移植、嵌入式数据库移植、嵌入式 Web 服务器移植和嵌入式 JVM 移植等方面介绍嵌入式应用程序的实践过程。

勘误和支持

由于作者的水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。为此，特意创建一个在线支持与应急方案的二级站点 http://book.blendercn.org。你可以将书中的错误发布在 Bug 勘误表页面中，同时如果你遇到任何问题，也可以访问 Q&A 页面，我将尽量在线上为你提供最满意的解答。书中的全部资源文件除可以从华章网站（www.hzbook.com）下载外，还可以从我创建的这个站点下载，我也会将相应功能的更新及时更正出来。如果你有更多的宝贵意见，也欢迎发送邮件至邮箱 lightfather@gmail.com，期待能够得到你们的真挚反馈。

致谢

首先要感谢伟大的 Linux 创始人 Linus Torvalds，是他开创了一款影响整个嵌入式行业的操作系统。

感谢成都国嵌信息技术有限公司，尤其要感谢国嵌的谢伟老师，是他建议我将自己的嵌入式移植实战经验编写成书，他不仅是一位知识渊博的老师，也是一位实战经验丰富的嵌入式工程师。在编写本书的过程中，得到了谢伟老师很多宝贵的建议。

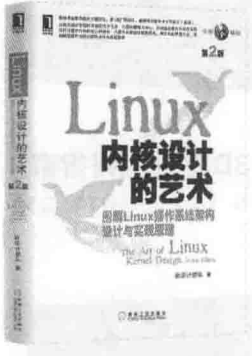
感谢全国众多的国嵌学员，我们曾经常在一起讨论本书应该以什么形式编写，正是这些源自潜在读者的宝贵建议，促成了本书以移植实践为目的的写作动机。

感谢机械工业出版社华章公司的编辑姜影老师，在这一年多的时间中始终支持我的写作，她的鼓励和帮助引导我能顺利完成全部书稿。

谨以此书献给众多热爱嵌入式 Linux 的朋友们！

范展源

推荐阅读



Linux内核设计与实现（原书第3版）

Linux内核开发人员Robert Love的力作，畅销多年的经典著作

Linux内核探秘：深入解析文件系统和设备驱动的架构与设计

腾讯顶级Linux系统专家和存储系统专家10年经验结晶

深度探索Linux操作系统：系统构建和原理解析

百度核心系统部门资深专家力作，Linux操作系统领域的里程碑作品

Linux内核设计的艺术：图解Linux操作系统架构设计与实现原理（第2版）

中国首部将版权输出到美国的计算机图书，中美两国取得骄人成绩

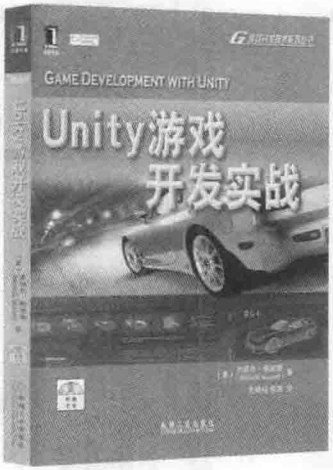
Linux内核精髓：精通Linux内核必会的75个绝技

日本多位一线内核技术专家的经验和智慧结晶

UNIX/Linux程序设计教程

UNIX/Linux权威著作，多所高校选定为教材

推荐阅读

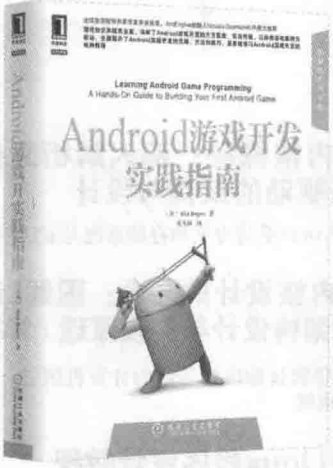


Unity游戏开发实战

作者：Michelle Menard ISBN：978-7-111-37719-1 定价：69.00元

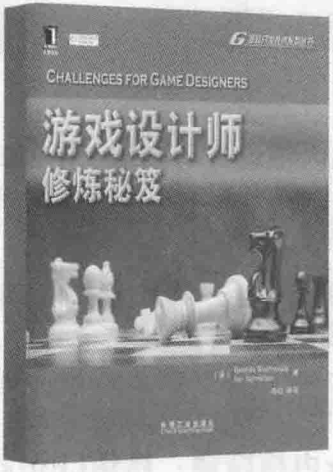
iPhone 3D游戏编程指南

作者：Jeremy Alessi ISBN：978-7-111-34937-2 定价：55.00元



Android游戏开发实践指南

作者：Rick Rogers ISBN：978-7-111-39154-8 定价：79.00元



游戏设计师修炼秘笈

作者：Brenda Brathwaite ISBN：978-7-111-34295-3 定价：49.00元

目　录 Contents

前言

绪论篇

第 1 章　嵌入式系统架构与移植环境搭建 ····· 2

1.1　嵌入式系统硬件架构 ····· 2
1.1.1　微处理器 ····· 3
1.1.2　总线 ····· 4
1.1.3　存储器 ····· 5
1.2　嵌入式系统软件架构 ····· 6
1.3　嵌入式 Linux 移植环境搭建 ····· 7
1.3.1　Ubuntu 开发平台 ····· 7
1.3.2　搭建交叉编译环境 ····· 8
1.3.3　获取内核 ····· 9
1.3.4　获取启动加载器 ····· 9
1.3.5　配置必要服务 ····· 9
1.3.6　PuTTY 的安装和配置 ····· 12
1.4　本章小结 ····· 12

u-boot 移植篇

第 2 章　u-boot 工程与编译系统 ····· 14

2.1　u-boot 介绍 ····· 14
2.1.1　u-boot 工程简介 ····· 15
2.1.2　u-boot 源码结构 ····· 15
2.1.3　u-boot 的配置编译 ····· 16
2.2　u-boot 常用命令与测试 ····· 18
2.2.1　获取帮助 ····· 18
2.2.2　环境变量相关命令 ····· 19
2.2.3　网络命令 ····· 20
2.2.4　Nand Flash 操作命令 ····· 21
2.2.5　内存 / 寄存器相关命令 ····· 22
2.2.6　系统引导命令 ····· 24
2.3　u-boot 编译过程分析 ····· 25
2.3.1　主机构建环境配置过程 ····· 25
2.3.2　目标机相关配置过程 ····· 27
2.3.3　make 命令执行过程 ····· 31
2.4　本章小结 ····· 40

第 3 章　u-boot 启动流程分析 ····· 41

3.1　u-boot 启动第一阶段流程 ····· 41
3.1.1　设置异常向量 ····· 42
3.1.2　CPU 进入 SVC 模式 ····· 43
3.1.3　设置控制寄存器地址 ····· 43
3.1.4　关闭看门狗 ····· 43
3.1.5　屏蔽中断 ····· 43
3.1.6　设置 MPLLCON、UPLLCON 和 CLKDIVN ····· 44
3.1.7　关闭 MMU 和 cache ····· 45

3.1.8 初始化存储控制器 ····· 46
3.1.9 复制 u-boot 第二阶段代码到 RAM ····· 47
3.1.10 设置栈 ····· 48
3.1.11 清除 BSS 段 ····· 48
3.1.12 跳转到第二阶段代码入口 ····· 48
3.2 u-boot 启动第二阶段代码分析 ····· 49
3.2.1 gd_t 结构体 ····· 49
3.2.2 bd_t 结构体 ····· 50
3.2.3 init_sequence 数组 ····· 50
3.2.4 start_armboot() 顺序分析 ····· 51
3.2.5 main_loop 函数分析 ····· 52
3.3 本章小结 ····· 56

第 4 章 ARM9/S3C2440 u-boot 移植实战 ····· 57

4.1 移植准备工作 ····· 57
4.1.1 开发环境介绍 ····· 57
4.1.2 删减 u-boot 文件 ····· 58
4.1.3 建立 My2440 配置 ····· 59
4.1.4 配置和编译 u-boot ····· 60
4.2 u-boot 芯片级移植 ····· 61
4.2.1 启动代码结构优化 ····· 61
4.2.2 系统时钟移植 ····· 65
4.2.3 存储器控制器设置 ····· 68
4.2.4 在 u-boot 工程中全面支持 CONFIG_S3C2440 宏配置 ····· 69
4.3 u-boot 调试方法探索 ····· 70
4.3.1 通过 LED 指示运行状态 ····· 70
4.3.2 在第一阶段的代码中添加打印调试方法 ····· 72
4.3.3 在内存中加载和运行 u-boot ····· 78
4.4 Nor Flash 驱动移植 ····· 78
4.4.1 Nor Flash 的工作模式 ····· 78
4.4.2 Nor Flash 的存储结构 ····· 79
4.4.3 Nor Flash 的硬件连接 ····· 79
4.4.4 Nor Flash 的操作方法 ····· 80
4.4.5 Nor Flash 驱动分析 ····· 83
4.4.6 Nor Flash 驱动移植 ····· 86
4.4.7 Nor Flash 命令测试 ····· 87
4.4.8 完善 u-boot 的命令行功能 ····· 88
4.5 DM9000 驱动移植 ····· 89
4.5.1 DM9000 网卡端口访问 ····· 90
4.5.2 DM9000 网卡时序分析 ····· 90
4.5.3 DM9000 网卡驱动分析 ····· 93
4.5.4 DM9000 网卡驱动移植 ····· 100
4.5.5 网卡驱动测试 ····· 103
4.5.6 通过 TFTP 下载程序到内存运行 ····· 103
4.6 u-boot 启动内核 ····· 105
4.6.1 ARM 架构的 Linux 内核启动 ····· 105
4.6.2 内核标记列表 ····· 106
4.6.3 u-boot 启动命令分析 1——go 命令 ····· 109
4.6.4 u-boot 启动命令分析 2——bootm 命令 ····· 110
4.6.5 u-boot 启动命令的配置与移植 ····· 115
4.7 Nand Flash 与驱动移植 ····· 118
4.7.1 Nand Flash 启动原理 ····· 118
4.7.2 Nand Flash 硬件特性 ····· 119
4.7.3 Linux MTD 子系统 ····· 121
4.7.4 Nand Flash 初始化流程分析 ····· 123
4.7.5 Nand Flash 命令实现分析 ····· 125
4.7.6 页读取操作详解 ····· 127
4.7.7 Nor Flash 和 Nand Flash 启动自动判断 ····· 132
4.7.8 Nand Flash 拷贝代码实现 ····· 132
4.7.9 Nand Flash 板级驱动移植 ····· 136
4.7.10 Nand Flash 命令测试 ····· 140
4.8 YAFFS 文件系统移植 ····· 142
4.8.1 YAFFS 文件系统 ····· 142
4.8.2 YAFFS 在 Nand Flash 上的

存储结构 …… 142
4.8.3 YAFFS 在内存中的组织结构 …… 143
4.8.4 在 u-boot 中添加对烧写 YAFFS 镜像的支持 …… 144
4.8.5 YAFFS 文件系统镜像制作 …… 150
4.8.6 YAFFS 的烧写与测试 …… 152
4.9 UBIFS 文件系统移植 …… 153
4.9.1 UBI 层 …… 153
4.9.2 UBIFS 介绍 …… 155
4.9.3 在 u-boot 中添加对 UBIFS 的支持 …… 156
4.9.4 制作 UBIFS 文件系统镜像 …… 157
4.9.5 UBIFS 的烧写与测试 …… 158
4.10 SD 卡驱动移植 …… 162
4.10.1 MMC/SD/SDIO 介绍 …… 162
4.10.2 SD/MMC 协议 …… 162
4.10.3 S3C2440 SDI 控制器操作 …… 166
4.10.4 SD 卡驱动分析 …… 168
4.10.5 SD 卡驱动移植 …… 173
4.11 USB 驱动移植 …… 176
4.11.1 USB 概述 …… 176
4.11.2 USB 系统架构 …… 177
4.11.3 USB 的通信方法 …… 180
4.11.4 USB 的描述符 …… 184
4.11.5 USB 设备请求 …… 188
4.11.6 USB 设备枚举 …… 191
4.11.7 S3C2440 USB 设备控制器 …… 195
4.11.8 u-boot USB 设备控制器驱动分析 …… 196
4.11.9 USB 设备驱动移植 …… 206
4.11.10 USB 驱动移植 …… 210
4.11.11 USB 功能测试 …… 214
4.12 u-boot 一键式菜单实现 …… 215
4.12.1 一键式菜单需求分析 …… 215
4.12.2 一键式菜单测试步骤 …… 216
4.12.3 一键式菜单源码分析 …… 220
4.13 本章小结 …… 224

第 5 章 ARM11/S3C6410 u-boot 移植实战 …… 225

5.1 移植准备工作 …… 225
5.1.1 开发环境 …… 225
5.1.2 删减 u-boot 文件 …… 226
5.1.3 建立 My6410 配置 …… 227
5.1.4 配置和编译 u-boot …… 228
5.2 u-boot 芯片级移植 …… 229
5.2.1 修改第一阶段启动代码 start.S …… 229
5.2.2 板级底层初始化文件 lowlevel_init.S 移植 …… 233
5.2.3 时钟初始化函数移植 …… 239
5.2.4 内存初始化函数实现 …… 243
5.2.5 Nand Flash 复制代码实现 …… 246
5.2.6 SD 卡复制代码实现 …… 250
5.2.7 底层调试方法探索 …… 254
5.2.8 完善 My6410 的板级配置 …… 258
5.2.9 烧写与测试 …… 265
5.3 DM9000 驱动移植 …… 267
5.3.1 DM9000 网卡端口访问 …… 267
5.3.2 DM9000 网卡时序分析 …… 268
5.3.3 DM9000 网卡驱动移植 …… 270
5.3.4 网卡驱动测试 …… 274
5.4 u-boot 启动内核 …… 274
5.4.1 u-boot 启动命令的配置与移植 …… 275
5.4.2 修改 go 命令以支持 zImage 格式镜像的启动 …… 277
5.5 Nand Flash 驱动移植 …… 278
5.5.1 将 u-boot 下载到内存中运行 …… 278
5.5.2 Nand Flash 驱动移植 …… 279
5.5.3 Nand Flash 命令测试 …… 289
5.6 YAFFS 文件系统移植 …… 291

5.6.1 在 u-boot 中添加对烧写 YAFFS 镜像的支持 ····· 291
5.6.2 YAFFS 文件系统镜像制作 ····· 298
5.6.3 YAFFS 的烧写与测试 ····· 299
5.7 UBIFS 文件系统移植 ····· 300
5.7.1 在 u-boot 中添加对 UBIFS 的支持 ····· 300
5.7.2 UBIFS 文件系统镜像制作 ····· 302
5.7.3 UBIFS 的烧写与测试 ····· 303
5.8 SD 卡驱动移植 ····· 307
5.8.1 S3C6410 主机控制器操作 ····· 307
5.8.2 S3C6410 主机控制器操作序列 ··· 308
5.8.3 SD 卡驱动分析 ····· 310
5.8.4 SD 卡驱动移植 ····· 317
5.8.5 通过 SD 卡更新系统 ····· 319
5.9 USB 驱动移植 ····· 321
5.9.1 S3C6410 USB2.0 高速 OTG ····· 321
5.9.2 u-boot USB 设备控制器驱动分析 ····· 323
5.9.3 USB 设备驱动移植 ····· 332
5.9.4 USB 功能测试 ····· 335
5.10 本章小结 ····· 336

Linux 内核移植篇

第 6 章 Linux 内核工程与编译系统 ····· 338

6.1 Linux 内核架构 ····· 338
6.1.1 内核体系结构 ····· 338
6.1.2 内核组件 ····· 339
6.1.3 内核目录结构 ····· 340
6.2 Linux 内核的配置与编译 ····· 341
6.2.1 配置内核 ····· 341
6.2.2 编译内核 ····· 344
6.3 Linux 内核构建系统 ····· 345
6.3.1 内核配置过程 ····· 345
6.3.2 扩展内核代码 ····· 347
6.3.3 内核中的 Makefile ····· 348
6.3.4 内核中的 Kconfig ····· 349
6.4 内核调试技术 ····· 352
6.4.1 调试准备 ····· 352
6.4.2 内核调试配置选项 ····· 352
6.4.3 源码级别的调试接口 ····· 353
6.4.4 使用 printk() 打印调试信息 ····· 355
6.4.5 使用 strace 跟踪系统调用 ····· 357
6.4.6 使用 OOPS 调试系统故障 ····· 358
6.5 本章小结 ····· 360

第 7 章 Linux 内核启动流程分析 ····· 361

7.1 内核镜像生成 ····· 361
7.2 内核启动流程 1——汇编部分 ····· 362
7.2.1 内核启动代码入口 ····· 362
7.2.2 深入源码分析 ····· 363
7.2.3 汇编启动代码分析总结 ····· 378
7.3 内核启动流程 2——C 语言部分 ····· 378
7.3.1 start_kernel() 函数 ····· 379
7.3.2 rest_init() 函数 ····· 388
7.3.3 kernel_init() 函数 ····· 390
7.3.4 init_post() 函数 ····· 391
7.4 内核启动流程 3——Busybox 的 init 进程 ····· 393
7.4.1 init 进程启动流程 ····· 393
7.4.2 添加初始化活动 ····· 394
7.4.3 执行初始化活动 ····· 395
7.5 本章小结 ····· 396

第 8 章 Linux 移植准备及最小系统构建 ····· 397

8.1 移植准备工作 ····· 397
8.1.1 开发环境 ····· 397
8.1.2 删减 Linux 文件 ····· 398
8.1.3 建立 My2440 配置 ····· 400

8.1.4 建立 My6410 配置……403
8.1.5 编译测试……406
8.2 最小系统搭建……409
8.2.1 嵌入式根文件系统制作……409
8.2.2 安装 initramfs 根文件系统……412
8.3 本章小结……414

第 9 章 Linux 网卡驱动移植……415

9.1 Linux 网络子系统……415
9.2 核心数据结构……416
9.2.1 net_device 结构……416
9.2.2 sk_buff 结构……419
9.3 DM9000 网卡驱动分析……421
9.3.1 board_info 结构……422
9.3.2 dm9000_probe() 函数……423
9.3.3 dm9000_open() 函数……427
9.3.4 dm9000_start_xmit() 函数……427
9.3.5 数据包接收函数……428
9.3.6 数据包发送函数……429
9.3.7 中断处理函数……431
9.4 My2440 网卡驱动移植……432
9.4.1 添加 DM9000 的平台设备……432
9.4.2 在内核配置添加对 DM9000 的支持……434
9.5 My6410 网卡驱动移植……434
9.5.1 添加 DM9000 的平台设备……434
9.5.2 在内核配置中添加对网络子系统的支持……435
9.5.3 在内核配置添加对 DM9000 的支持……436
9.6 安装 NFS 根文件系统……436
9.6.1 在内核配置添加对 NFS 的支持……436
9.6.2 挂载 NFS 根文件系统……437
9.7 制作基于共享库的根文件系统……437
9.8 本章小结……439

第 10 章 Linux 混杂设备驱动……440

10.1 My2440 RTC 驱动移植……440
10.2 My6410 RTC 驱动移植……441
10.2.1 修改 RTC 驱动 rtc-s3c.c……441
10.2.2 完善对 6410 RTC 驱动的平台支持……445
10.2.3 在机器配置文件中添加 RTC 设备……448
10.2.4 在内核中配置 RTC……448
10.3 RTC 驱动测试……449
10.4 为 My2440 添加 ADC 和按键驱动……451
10.4.1 按键驱动分析……451
10.4.2 在内核中添加 ADC 和按键驱动……454
10.5 为 My6410 添加 ADC 驱动……457
10.6 本章小结……458

第 11 章 Linux I2C 驱动移植……459

11.1 I2C 协议概述……459
11.1.1 I2C 总线物理拓扑结构……459
11.1.2 I2C 通信协议……460
11.2 Linux I2C 子系统框架……461
11.3 I2C 驱动中的数据结构及操作……462
11.3.1 i2c_adapter 结构……462
11.3.2 i2c_algorithm 结构……464
11.3.3 i2c_msg 结构……464
11.3.4 i2c_driver 结构……465
11.3.5 i2c_client 结构……467
11.4 I2C 适配器的设备接口……468
11.4.1 i2cdev_open() 函数……471
11.4.2 i2cdev_ read() 函数……471
11.4.3 i2cdev_ioctl() 函数……472
11.5 S3C2440（6410）I2C 适配器驱动的实现……473
11.5.1 S3C2440 I2C platform 总线

匹配……474
11.5.2 S3C2440 I2C 总线驱动描述结构……474
11.5.3 probe 方法的实现……476
11.5.4 S3C2440 I2C 总线通信方法…477
11.6 S3C2440（6410）I2C 适配器驱动移植……480
11.6.1 添加 I2C 平台设备……480
11.6.2 在内核配置中支持 I2C 驱动……481
11.6.3 编写 I2C 总线驱动测试程序…482
11.7 S3C2440（6410）I2C 设备驱动的实现……484
11.7.1 At24 系列 I2C EEPROM 设备驱动的实现……484
11.7.2 传统只读 EEPROM 设备驱动的实现……486
11.8 I2C EEPROM 设备驱动移植……489
11.9 本章小结……490

第 12 章 Linux SPI 驱动移植……491

12.1 SPI 协议概述……491
12.1.1 SPI 总线物理拓扑结构……491
12.1.2 时钟极性和时钟相位……492
12.1.3 SPI 的优缺点……493
12.2 Linux SPI 子系统……493
12.3 SPI 驱动中的数据结构及操作……494
12.3.1 spi_master 结构……494
12.3.2 spi_driver 结构……495
12.3.3 spi_device 结构……496
12.3.4 spi_message 结构……497
12.3.5 spi_bitbang 结构……498
12.4 SPI 控制器的设备接口……500
12.5 S3C2440 SPI 控制器驱动的实现…503
12.5.1 S3C2440 SPI platform 总线匹配……503
12.5.2 S3C2440 SPI 控制器驱动描述结构……504
12.5.3 probe 方法的实现……504
12.5.4 S3C2440 SPI 总线通信方法…505
12.6 S3C6410 SPI 控制器驱动的实现…507
12.6.1 S3C6410 SPI 控制器驱动描述结构……507
12.6.2 probe 方法的实现……507
12.6.3 S3C6410 SPI 总线通信方法…509
12.7 S3C2440 SPI 控制器驱动移植……510
12.7.1 在机器配置文件中添加对 SPI 的支持……510
12.7.2 扩展 Kconfig……512
12.7.3 在内核配置中支持 SPI 驱动…513
12.7.4 SPI 驱动测试……513
12.8 S3C6410 SPI 控制器驱动移植……513
12.8.1 添加 6410 的 SPI 驱动……514
12.8.2 添加 SPI 平台设备……514
12.8.3 添加 6410 DMA 平台代码……516
12.8.4 添加 SPI、DMA 相关的时钟资源……518
12.8.5 在机器配置文件中添加对 SPI 的支持……519
12.8.6 SPI 驱动测试……521
12.9 S3C2440（S3C6410）SPI 协议驱动移植……521
12.9.1 AT25 系列 SPI EEPROM 协议驱动的实现……521
12.9.2 SPI EEPROM 设备驱动移植…523
12.10 本章小结……524

第 13 章 Nand Flash 驱动移植……525

13.1 Linux MTD 子系统……525
13.2 MTD 子系统中的数据结构及操作……526
13.2.1 mtd_info 结构……526

13.2.2 mtd_notifier 结构……528
13.2.3 mtd_part/mtd_partitions 结构……528
13.3 MTD 块设备实现分析……530
13.4 Nand Flash 驱动的实现……534
13.4.1 Nand Flash platform 总线匹配……534
13.4.2 S3C2440 Nand Flash 控制器驱动描述结构……534
13.4.3 probe 方法的实现……536
13.4.4 S3C2440 Nand Flash 读写方法分析……537
13.5 S3C2440 Nand Flash 控制器驱动移植……538
13.5.1 在机器配置文件中添加对 Nand Flash 的支持……538
13.5.2 完善 S3C2440 的 Nand Flash 驱动……540
13.5.3 在内核配置中支持 Nand Flash 驱动……541
13.6 S3C6410 Nand Flash 控制器驱动移植……541
13.6.1 添加 6410 Nand Flash 驱动……541
13.6.2 添加 Nand Flash 平台设备……542
13.6.3 在机器配置文件中添加对 Nand Flash 的支持……544
13.6.4 其他修改……545
13.7 YAFFS2 文件系统移植……547
13.7.1 将 YAFFS2 文件系统移植到 Linux 内核……548
13.7.2 利用 mtd-utils 烧写 YAFFS2 文件系统镜像……548
13.8 在内核中配置对 UBIFS 的支持……550
13.9 本章小结……550

第 14 章 SD/MMC 卡驱动移植……551

14.1 Linux MMC 子系统……551
14.2 MMC 子系统中的数据结构及操作……552
14.2.1 mmc_host 结构……552
14.2.2 mmc_card 结构……554
14.2.3 mmc_driver 结构……555
14.2.4 mmc_request 结构……556
14.3 卡探测过程分析……558
14.4 S3C2440 SD/MMC 主控制器驱动的实现……561
14.4.1 SD/MMC 主控制器驱动 platform 总线匹配……561
14.4.2 S3C2440 SD/MMC 主控制器驱动描述结构……561
14.4.3 probe 方法的实现……563
14.4.4 S3C2440 SD/MMC 主控制器请求处理分析……565
14.5 S3C6410 高速 MMC 控制器驱动的实现……568
14.5.1 SDHCI 驱动框架……569
14.5.2 SDHCI 主机卡探测过程……569
14.5.3 S3C6410 HSMMC 控制器驱动……570
14.5.4 S3C6410 HSMMC 控制器请求处理……572
14.6 S3C2440 SD/MMC 主机控制器驱动移植……572
14.6.1 在机器配置文件中添加对 SD/MMC 主控制器的支持……572
14.6.2 在内核配置中添加对 SD/MMC 主控制器驱动的支持……573
14.6.3 在内核配置中添加对中文字符集和 FAT 文件系统的支持……574
14.6.4 在文件系统中添加对热插拔事件处理的支持……574
14.6.5 SD/MMC 主控制器驱动测试……575
14.7 S3C6410 HSMMC 控制器驱动移植……576

14.7.1 添加 6410 Nand Flash 驱动 …… 576
14.7.2 在机器配置文件中添加对 HSMMC 的支持 …… 577
14.7.3 修改内核 SDHCI 驱动 …… 578
14.8 本章小结 …… 579

第 15 章 LCD 驱动移植 …… 580

15.1 LCD 硬件原理 …… 580
15.1.1 LCD 的硬件构成 …… 580
15.1.2 TFT 屏显示时序 …… 581
15.2 Linux 帧缓冲子系统 …… 582
15.3 帧缓冲子系统中的数据结构及操作 …… 583
15.3.1 fb_info 结构 …… 583
15.3.2 fb_var_screeninfo 结构和 fb_fix_screeninfo 结构 …… 587
15.3.3 fb_cmap 结构 …… 589
15.4 帧缓冲字符设备接口 …… 590
15.5 Linux 显示启动 Logo …… 594
15.6 S3C2440 帧缓冲驱动的实现 …… 596
15.6.1 S3C2440 LCD 控制器硬件描述 …… 596
15.6.2 驱动 platform 总线匹配 …… 597
15.6.3 S3C2440 帧缓冲驱动描述结构 …… 598
15.6.4 probe 方法实现 …… 599
15.7 S3C6410 帧缓冲驱动的实现 …… 601
15.8 S3C2440 帧缓冲驱动移植 …… 601
15.8.1 在板级初始化文件中添加对帧缓冲的支持 …… 601
15.8.2 修改 Makefile 和 Kconfig …… 605
15.8.3 在内核配置中添加对帧缓冲驱动的支持 …… 607
15.8.4 帧缓冲驱动测试 …… 608
15.9 S3C6410 帧缓冲驱动移植 …… 608
15.9.1 添加 6410 帧缓冲驱动 …… 608
15.9.2 在机器配置文件中添加对帧缓冲的支持 …… 609
15.9.3 帧缓冲驱动测试 …… 611
15.10 本章小结 …… 611

第 16 章 触摸屏驱动移植 …… 612

16.1 Linux 输入子系统 …… 612
16.2 输入子系统中的数据结构及操作 …… 613
16.2.1 input_dev 结构 …… 613
16.2.2 input_handler 结构 …… 616
16.2.3 input_handle 结构 …… 617
16.3 输入子系统核心层的实现 …… 618
16.4 通用事件处理驱动 …… 622
16.5 输入事件报告流程 …… 626
16.6 S3C2440（6410）触摸屏驱动的分析 …… 629
16.6.1 模块初始化函数的实现 …… 629
16.6.2 中断处理与事件上报 …… 630
16.7 S3C2440 触摸屏驱动移植与测试 …… 632
16.7.1 S3C2440 触屏驱动移植 …… 632
16.7.2 S3C2440 触屏驱动测试 …… 633
16.8 S3C6410 触摸屏驱动移植与测试 …… 634
16.8.1 添加 6410 的触摸屏驱动 …… 634
16.8.2 添加触屏平台设备 …… 635
16.8.3 在机器配置文件中添加对触屏的支持 …… 636
16.9 本章小结 …… 637

第 17 章 声卡驱动移植 …… 638

17.1 ALSA 体系架构 …… 638
17.1.1 ALSA 设备文件 …… 639
17.1.2 驱动代码的文件结构 …… 640
17.2 声卡描述结构 snd_card …… 640

17.3 PCM 功能子层……644
17.3.1 PCM 的概念……644
17.3.2 PCM 设备描述结构 snd_pcm…645
17.3.3 PCM 流与 PCM 子流……646
17.3.4 PCM 功能子层逻辑关系小结…651
17.3.5 PCM 设备文件的建立……652
17.3.6 PCM 设备文件的访问……654
17.4 声卡控制项……655
17.4.1 控制项创建……655
17.4.2 控制项回调函数……657
17.4.3 Control 设备建立……658
17.5 ASoC 声卡驱动架构……659
17.6 ASoC 架构中的 Machine 驱动……662
17.6.1 创建 ASoC 声卡平台设备……662
17.6.2 ASoC 声卡的平台驱动……665
17.7 ASoC 架构中的 Codec 驱动……666
17.7.1 Codec 的 DAI 和 PCM 的配置…666
17.7.2 Codec 的控制 IO……669
17.7.3 混音器和其他音频控制项……671
17.7.4 Codec 设备与 ASoC 声卡注册…673
17.8 ASoC 架构中的 Platform 驱动……676
17.8.1 CPU DAI 驱动……676
17.8.2 音频 DMA 驱动……677
17.8.3 创建音频 DMA 缓冲区……678
17.8.4 音频 DMA 的 PCM 操作……681
17.9 S3C2440+UDA1341 声卡驱动配置与测试……683
17.9.1 在机器配置文件中添加对声卡的支持……683
17.9.2 在内核配置中支持声卡驱动…684
17.9.3 应用层 alsa-lib 移植……685
17.9.4 编写 ALSA 应用程序……686
17.9.5 播放和录音测试……690
17.10 S3C6410+WM9714 声卡驱动移植……691
17.10.1 添加 6410 声卡驱动……691
17.10.2 在内核配置中支持声卡驱动……692
17.10.3 alsa-utils 工具集移植……692
17.11 本章小结……693

第 18 章 USB 驱动移植……694
18.1 USB 子系统架构……694
18.2 USB 驱动中的描述符结构……695
18.3 USB 主机驱动……695
18.3.1 主机控制器驱动……695
18.3.2 OHCI 主机控制器驱动……698
18.4 S3C2440/S3C6410 USB 主机驱动的实现……700
18.5 USB 设备驱动……701
18.5.1 USB 设备驱动描述结构……702
18.5.2 USB 请求块 URB……703
18.5.3 URB 的处理流程……705
18.5.4 usb_bulk_msg() 和 usb_control_msg()……708
18.5.5 探测和断开函数……709
18.6 USB 骨架程序……710
18.7 USB 设备驱动实例……718
18.7.1 DNW 驱动的实现……718
18.7.2 USB 键盘驱动的实现……720
18.8 本章小结……728

应用程序移植篇

第 19 章 嵌入式 Qt 移植……730
19.1 Qt 开发环境搭建与使用……730
19.1.1 Qt SDK 的下载与安装……730
19.1.2 第一个 Qt 程序……732
19.1.3 利用 Qt Creator 建立一个工程…733
19.2 Qt 的功能模块与裁剪……735
19.2.1 Qt 模块的构成……735