

深入浅出嵌入式底层软件开发



[深入浅出嵌入式底层软件开发 下载链接1](#)

著者:杨铸//唐攀

出版者:北京航空航天大学

出版时间:2011-5

装帧:

isbn:9787512403826

《深入浅出: 嵌入式底层软件开发》内容简介: 第1篇 (第1~3章) 以ARM CPU及其汇编语言为背景, 深入浅出地讲解软件是如何控制硬件的。第2篇 (第4~7章) 以S3C2440开发板为硬件, 讲解制作嵌入式Linux系统的完整过程, 具有很强的实用性和可操作性。第3篇 (第8~11章) 以尽可能通俗的方式讲解Linux设备驱动中涉及的复杂

理论，在此基础上，以驱动程序的2个接口（向下与硬件接口、向上与内核子系统接口）为突破口，由浅入深地编写和剖析了嵌入式真实典型硬件（LED灯、按键、网卡、LCD、触摸屏）的驱动程序。所有的驱动程序均能在实际硬件上正常运行，帮助读者揭开Linux驱动的神秘面纱。第12章以实战的方式，深入浅出地剖析了嵌入式Linux系统中最重要的2类驱动（LCD和触摸屏）的代码实现。并简介了块设备驱动和usb设备驱动。本书适合硬件工程师、软件工程师、嵌入式软件的从业人员、教授嵌入式软件开发课程的老师、意欲从事嵌入式软件开发工作的大学生阅读。

作者介绍：

杨铸

教育背景：电子科技大学通信与信息工程工学硕士；北京邮电大学工学学士学位，专业为计算机软件。重大经历：作为主要组织人员，组织完成了中国移动四川分公司最早的短信系统平台的建设；早期专注于Microsoft的OS和Database的技术研究与职业培训，是微软在中国的早期MCT之一，MCP号1694198；参与完成国家863计划项目——信息安全产品演示和验证平台的子课题并与合作公司完成产品化开发——内部网络监管系统；中后期专注于嵌入式软件开发的研究与职业培训，精于ARM体系结构与嵌入式Linux。

目录: 第1篇 ARM体系结构与编程 第1章 ARM汇编编程基础 1.1 ARM CPU寄存器 1.1.1 普通寄存器R0~R15 1.1.2 状态寄存器CPSR与SPSR 1.1.3 流水线对PC的值的影响 1.2 基本寻址方式与基本指令 1.2.1 最常见寻址方式精解 1.2.2 最常见指令精解 1.3 ARM汇编伪操作 1.3.1 汇编伪操作在汇编程序中的使用范例 1.3.2 最常见汇编伪操作精解 1.3.3 汇编伪操作列表 1.4 ADS开发环境的使用 1.4.1 在ADS中进行裸机程序的编辑、编译、运行 1.4.2 在AXD中进行裸机程序调试的方法与步骤 1.5 RealViewMDK开发环境的使用 1.5.1 在MDK开发环境下编写裸机程序 1.5.2 MDK调试裸机程序的方法与步骤 1.6 其他常见寻址模式与常见指令 1.6.1 其他常见寻址模式 1.6.2 其他常见指令 第2章 ARM编程进阶 2.1 ARM汇编伪指令 2.1.1 精解ldr伪指令 2.1.2 精解adr 2.1.3 精解adrl伪指令 2.1.4 nop伪指令 2.2 ATPCS与混合编程 2.2.1 ATPCS规则精解 2.2.2 精解C和ARM汇编程序间的相互调用 2.3 裸机硬件的控制方法与例程 2.3.1 建立真实硬件的开发和调试环境 2.3.2 软件控制（驱动）硬件的编程原理 2.3.3 裸机硬件控制程序实例 2.3.4 启动例程 2.4 看门狗定时器 2.4.1 看门狗定时器的用途 2.4.2 看门狗工作原理 2.4.3 看门狗实验 2.5 系统时钟 2.5.1 系统工作时钟频率 2.5.2 时钟驱动实验 2.6 SDRAM内存 2.6.1 3C2440存储器地址段（Bank） 2.6.2 SDRAM内存工作原理 2.6.3 SDRAM的读操作 2.6.4 SDRAM预充电操作 2.6.5 SDRAM突发操作 2.6.6 SDRAM写操作 2.6.7 SDRAM的刷新 2.6.8 内存驱动实验 2.7 UART串口 2.7.1 同步通信和异步通信 2.7.2 数据的串行和并行通信方式 2.7.3 数据通信传输模式 2.7.4 S3C2440UART、控制器 2.7.5 S3C2440UART串口工作原理 2.7.6 UART、串口驱动实验 第3章 ARM体系结构 3.1 ARM处理器工作模式 3.1.1 ARM处理器不同模式下的寄存器 3.1.2 ARM处理器模式切换（含MRS、MSR指令） 3.2 ARM处理器异常处理 3.2.1 异常分类 3.2.2 异常发生时的硬件操作 3.2.3 异常返回地址 3.2.4 异常向量表 3.2.5 异常处理的返回 3.3 S3C2440系统中断 3.3.1 中断的产生-中断源 3.3.2 中断优先级 3.3.3 中断控制器相关寄存器 3.3.4 系统中断流程 3.3.5 按键控制LED灯实验 3.4 semihosting与硬件重定向 3.4.1 semihosting半主机调试 3.4.2 硬件重定向 3.5 系统调用与软件中断SWI的实现 3.5.1 系统调用 3.5.2 软件中断 3.5.3 软件中断处理 3.5.4 LED系统调用实验 3.6 进程切换的实现 3.6.1 进程 3.6.2 进程控制块PCB 3.6.3 进程创建 3.6.4 进程队列 3.6.5 进程调度 3.6.6 上下文切换 3.7 MMU与内存保护的实现 3.7.1 存储管理单元MMU 3.7.2 cache 3.7.3 CPI5协处理器 3.8 实战：小型多任务操作系统miniOS的实现 3.8.1 miniOS代码分析 3.8.2 miniOS应用程序接口 3.8.3 miniOS应用程序系统调用接口 第2篇 嵌入式Linux系统建构 第4章 嵌入式Linux软件开发环境搭建 4.1 体验嵌入式Linux系统 4.2 Linux操作系统安装 4.2.1 在Windows上安装虚拟机 4.2.2 在虚拟机上安装Linux操作系统ubuntu9.10 4.3

在ubuntu9.10中安装基本的开发环境 4.4 ubuntu9.10上网络服务的安装与配置 4.4.1
设置vmware网络 4.4.2 安装、配置和使用FTP服务 4.4.3 安装、配置NFS服务 第5章
建构BootLoader 5.1 准备工作 5.1.1 嵌入式Linux系统概述 5.1.2 构建交叉编译工具链
5.1.3 BootLoader概述 5.2 深入剖析u-boot代码 5.2.1
安装和使用源代码阅读工具SourceInsight 5.2.2 u-boot的编译初步 5.2.3
分析u-boot的第一阶段代码 (cpu/arm920t/start.S) 5.2.4 分析u-boot的第二阶段代码
5.2.5 继续移植、编译u-boot.. 5.2.6 u-boot常用命令使用简介 5.2.7
u-boot命令实现框架的分析 5.2.8 u-boot引导Linux操作系统的过程分析 5.2.9
让u-boot支持从USBslave接口获得数据 第6章 建构嵌入式Linux内核 6.1 Linux内核简介
6.1.1 Linux内核版本历史 6.1.2 内核源码目录结构 6.1.3 Linux内核构造系统简介 6.2
移植、裁减及配置Linux内核到S3C2440开发板 6.2.1 体验Linux内核配置、编译与使用
6.2.2 为S3C2440移植内核 6.2.3 配置并裁减内核 6.2.4 运行内核并验证内核被配置的功能
6.3 内核Kconfig与Makefile文件分析 6.3.1 内核构造系统简介 6.3.2 Kconfig文件精解
6.3.3.config文件说明 6.3.4 Makefile文件精解 6.3.5
实战：修改Kconfig和Makefile，完成向内核中添加新的功能组件——网卡、声卡、LCD
、触摸屏驱动 第7章 建构嵌入式Linux文件系统375 7.1 嵌入式Linux文件系统简介 7.1.1
嵌入式文件系统概述 7.1.2 Mrl、D设备与Flash文件系统简介 7.1.3
嵌入式Linux系统中的tmpfs文件系统 7.2 详解制作根文件系统 7.2.1 FHS标准介绍 7.2.2
编译/安装busybox，生成/1bin、/sbin、/usr/bin、/usr/sbin目录 7.2.3
利用交叉编译工具链，构建/lib目录 7.2.4 手工构建/etc目录 7.2.5
手工构建最简化的/dev目录 7.2.6
使用启动脚本完成/proc、/sys、/dev、/tmp、/var等目录的完整构建 7.2.7
制作根文件系统的jffs2映像文件 7.3 建构嵌入式Linux应用程序系统. 7.3.1
辅助处理工具的移植 7.3.2 MP3播放器madplay的移植 7.3.3
主要网络服务器的移植与使用 7.3.4 数据库程序的移植与使用 7.4 建构GUI系统 7.4.1
移植tslib库 7.4.2 移植qtopia第3篇 Linux驱动程序开发 第8章 Linux驱动程序开发基础
8.1 Linux设备驱动程序简介 8.1.1 设备驱动分类和内核模块 8.1.2 设备文件和设备驱动
8.1.3 内核模块的编译和使用 8.2 字符设备驱动基本编程 8.2.1 字符设备驱动体验 8.2.2
实现字符设备驱动的工作 8.3 驱动程序中的并发控制方法 8.3.1 并发控制原理简介 8.3.2
信号量的编程实战 8.3.3 自旋锁的编程实战 8.3.4 Linux内核提供的其他并发控制方法 8.4
驱动程序中的阻塞与非阻塞编程 8.4.1 体验阻塞I/O 8.4.2 如何在驱动程序中实现阻塞I/O
8.4.3 体验非阻塞I/O 8.4.4 如何在驱动程序中实现非阻塞I/O 8.5
字符设备驱动程序对一些高级特性的实现 8.5.1 non-seekable的实现 8.5.2 select的实现
第9章 Linux字符设备驱动开发实战 9.1 I/O内存与硬件通信 9.1.1 驱动中的内存分配 9.1.2
使用I/O端口地址空间与硬件进行通信的内核API介绍 9.1.3
使用I/O内存地址空间与硬件进行通信的内核API介绍 9.1.4
通过I/O内存驱动硬件的实战——LED灯驱动 9.1.5 驱动程序对ioctl的规范实现 9.2
内核misc设备架构分析 9.2.1 定义全局变量 9.2.2 注册主设备号为10的misc设备 9.2.3
导出内核API——misc_register函数 9.2.4 实施“乾坤大挪移”的misc设备open函数
9.2.5 导出内核API——misc_deregister函数 9.3 Watchdog驱动 9.3.1 相关概念 9.3.2
watchdog硬件结构分析 9.3.3 Watchdog驱动的初始化和卸载 9.3.4
探测函数watchdog_probe的实现 9.3.5 实现misc设备中对设备文件的操作 9.3.6
Watchdog平台驱动的设备移除、挂起和恢复接口函数的实现 9.3.7 测试Watchdog驱动
9.4 内核编码规范与风格 9.4.1 缩进、长行、{}与空格的使用规范 9.4.2 变量和函数 9.4.3
注释、macros和enums 9.4.4 快乐使用内核提供的实现常用功能的宏 第10章
Linux驱动中的中断编程 10.1 驱动程序调测方法与技巧 10.1.1 利用printf 10.1.2
详解OOP消息 10.1.3 利用strace 10.1.4 利用内核内置的hacking选项 10.1.5
其他调测方法简介 10.2 驱动程序中的中断处理 10.2.1 中断简述 10.2.2
驱动程序中进行中断处理涉及的最基本的内核API 10.2.3
驱动程序进行中断处理的实例代码分析 10.2.4 其他关于中断的内核APIj 10.3
内核时间与内核定时器 10.3.1 内核中如何记录时间 10.3.2 内核定时器API 10.3.3
内核定时器与内核时间的应用案例——按键消抖 10.3.4 如何在内核中实现延时 10.4
中断顶半部与底半部 10.4.1 区分和使用中断顶半部与底半部的原因 10.4.2
tasklet机制与编程实例 10.4.3 workqueue机制与编程实例 10.4.4

tasklet与workqueue的区别和不同应用环境总结 10.5

Linux中断处理系统的架构与共享中断 10.5.1

裸机程序中的中断编程与有操作系统下的中断编程的区别 10.5.2

Linux中断处理系统的架构 10.5.3 关于共享中断的说明 10.5.4 共享中断实例 第11章

Linux网络设备驱动开发实战547 11.1 网络设备驱动基础 11.1.1 体验网卡驱动 11.1.2

网卡驱动的基本知识——2个结构体和5个函数 11.1.3 虚拟网卡snnull驱动代码分析 11.1.4

网卡驱动的编写主要内容总结 11.2 网络设备驱动实例——cs8900 11.2.1

虚拟网卡驱动与真实网卡驱动的主要区别 11.2.2 真实网卡驱动的整体框架分析 11.2.3

驱动中关于cs8900硬件操作的探讨 第12章 其他重要设备驱动开发实战 12.1

块设备驱动初步（以ramdisk为例） 12.1.1 体验块设备驱动 12.1.2 块设备驱动框架介绍

12.1.3 块设备的简单读写实现代码分析 12.1.4 块设备的高效读写实现代码分析 12.1.5

块设备的其他操作接口fops 12.2 LCD驱动 12.2.1 LCD裸机驱动 12.2.2

帧缓冲设备驱动框架结构 12.2.3 LCD驱动实例代码 12.2.4

LCD驱动代码的主干结构的总结 12.2.5 测试LCD驱动程序 12.3 触摸屏驱动 12.3.1

触摸屏裸机驱动 12.3.2 Linux输入子系统 12.3.3 Linux下触摸屏驱动的实现步骤 12.3.4

测试触摸屏驱动程序 12.4 USB驱动初步 12.4.1 Linux下4种USB驱动简介与功能体验

12.4.2 USB接口与规范 12.4.3 USB设备驱动基本知识 12.4.4 USB设备驱动实例参考文献

· · · · · (收起)

[深入浅出嵌入式底层软件开发](#) [下载链接1](#)

标签

嵌入式

嵌入式驱动开发

linux

底层软件

博客藏经阁

arm

嵌入式，

驱动

评论

[深入浅出嵌入式底层软件开发 下载链接1](#)

书评

[深入浅出嵌入式底层软件开发 下载链接1](#)