

图书基本信息

书名：<<深入理解嵌入式Linux设备驱动程序>>

13位ISBN编号：9787121177194

10位ISBN编号：7121177196

出版时间：2012-8

出版时间：电子工业出版社

作者：曹国辉

页数：268

字数：420000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

内容概要

本书详细阐述了嵌入式Linux设备驱动程序基本理论及开发方法。详细讲解了嵌入式系统的硬件体系架构，Linux系统的引导原理、启动过程，Linux设备驱动模型及基本理论，具体的硬件设备驱动程序的源码分析。在分析具体设备驱动程序源码的基础上，详细讲解了具体设备驱动程序的设计和移植方法。

书籍目录

第1章 嵌入式arm系统开发基础

1.1 嵌入式arm系统开发概述

1.1.1 arm系统可执行映像文件格式

1.1.2 arm开发调试工具

1.1.3 加载地址和运行地址

1.2 嵌入式arm系统的启动代码分析

1.2.1 arm上电启动概述

1.2.2 arm上电初始化启动代码分析

1.3 嵌入式arm系统的中断系统

1.3.1 arm中断机制代码分析

1.3.2 arm中断服务处理程序的响应过程

1.4 按键中断实验

第2章 嵌入式linux设备驱动程序开发环境的构建

2.1 安装虚拟机软件vmware 6.0.2

2.2 新建虚拟机

2.3 安装linux操作系统ubuntu

2.4 安装vmware tools工具软件

2.5 网络配置

2.6 设置软件源服务器

2.7 安装libncurses5-dev软件包

2.8 安装交叉编译器arm-linux-gcc

第3章 嵌入式linux内核的裁剪与编译

3.1 嵌入式linux内核的本质

3.2 嵌入式linux内核源码的组织

3.3 嵌入式linux内核的移植与裁剪

3.4 嵌入式linux内核配置、编译的基本原理

3.5 构建嵌入式linux根文件系统

第4章 嵌入式系统bootloader代码分析与移植

4.1 嵌入式系统bootloader功能概述

4.1.1 嵌入式linux系统的软件组成及分布

4.1.2 嵌入式linux中为什么要有bootloader

4.1.3 bootloader的功能和选择

4.2 u-boot源码分析

4.2.1 u-boot源码文件的组成及配置编译

4.2.2 u-boot的执行过程及linux内核加载流程

4.2.3 start.s源码文件分析

4.2.4 board.c源码文件分析

4.3 u-boot中的环境变量

4.4 linux系统的加载过程

4.4.1 u-boot命令执行过程

4.4.2 bootm的执行流程

4.5 u-boot向linux传递参数的过程

4.6 u-boot的移植

第5章 嵌入式linux内核模块与字符设备驱动

5.1 嵌入式linux内核模块

- 5.1.1 嵌入式linux内核模块的概念
- 5.1.2 嵌入式linux内核模块编程
- 5.1.3 嵌入式linux内核模块的编译
- 5.1.4 嵌入式linux内核模块的安装与卸载
- 5.1.5 直接编译嵌入式linux内核模块到内核
- 5.2 嵌入式linux字符设备驱动
 - 5.2.1 嵌入式linux设备驱动程序的概念
 - 5.2.2 嵌入式linux设备管理机制
 - 5.2.3 嵌入式linux字符设备驱动程序的主要数据结构分析
 - 5.2.4 嵌入式linux字符设备驱动的工作原理
 - 5.2.5 嵌入式linux字符设备驱动程序设计实例
- 第6章 嵌入式linux系统的内存管理
 - 6.1 虚拟地址和物理地址概述
 - 6.2 虚拟地址到物理地址转换的基本原理
 - 6.3 基于arm s3c2440的gpio端口地址映射实验
 - 6.3.1 问题描述
 - 6.3.2 分析与思路
 - 6.4 linux内核中内存分配和释放函数的用法
- 第7章 嵌入式linux设备驱动开发的核心技术
 - 7.1 嵌入式linux中断处理和定时器
 - 7.1.1 嵌入式linux系统中断服务程序的编写
 - 7.1.2 嵌入式linux系统硬件定时器的使用
 - 7.1.3 中断的下半部分
 - 7.1.4 嵌入式linux软件定时器的使用
 - 7.2 嵌入式linux设备驱动程序中的并发及并发控制
 - 7.2.1 并发的概念
 - 7.2.2 嵌入式linux设备驱动程序中的并发控制方式
 - 7.2.3 信号量与自旋锁的使用场景
 - 7.3 嵌入式linux设备驱动中的阻塞与非阻塞
 - 7.3.1 概述
 - 7.3.2 linux设备驱动程序中阻塞的工作原理
 - 7.3.3 进程阻塞访问设备的基本原理
 - 7.3.4 linux设备驱动程序中的阻塞编程
 - 7.4 嵌入式linux设备驱动中的异步通知编程
 - 7.4.1 概述
 - 7.4.2 linux异步通知工作原理
 - 7.4.3 linux异步通知应用的编程方法
 - 7.4.4 linux异步通知驱动的编程方法
 - 7.5 嵌入式linux设备驱动中的轮询操作
 - 7.5.1 概述
 - 7.5.2 linux设备驱动轮询操作的工作原理及源码分析
 - 7.5.3 linux轮询操作的应用层编程
 - 7.5.4 linux轮询操作的驱动层编程
- 第8章 嵌入式linux平台设备驱动程序开发
 - 8.1 linux设备和设备驱动模型
 - 8.1.1 linux内核中的bus (总线)
 - 8.1.2 linux内核中的设备

- 8.1.3 linux内核中的设备驱动
- 8.2 linux平台设备驱动程序开发过程
- 8.3 嵌入式linux系统中利用mdev自动创建设备文件节点
- 第9章 嵌入式linux lcd屏驱动程序设计
 - 9.1 lcd屏的工作原理概述
 - 9.2 lcd屏硬件原理及驱动程序设计
 - 9.2.1 tft lcd屏的显示原理
 - 9.2.2 s3c2440 lcd控制器tft lcd的控制时序分析
 - 9.2.3 s3c2440 lcd控制器显示的数据格式
 - 9.2.4 s3c2440 lcd控制器的显示数据流程
 - 9.2.5 vbpd、vfpd、vspw和hbpd、hfpd、hspw的设置
 - 9.3 嵌入式linux lcd屏驱动程序框架
 - 9.4 嵌入式linux lcd屏驱动源码分析
 - 9.4.1 s3c2410fb.c源码分析
 - 9.4.2 lcd屏linux驱动主要数据结构
 - 9.4.3 probe函数处理流程及源码分析
 - 9.4.4 fbmem.c源码分析
 - 9.5 嵌入式linux lcd屏驱动的移植
- 第10章 嵌入式linux触摸屏驱动程序设计
 - 10.1 触摸屏工作原理概述
 - 10.2 s3c2440触摸屏接口及硬件驱动程序设计
 - 10.2.1 s3c2440触摸屏控制器接口
 - 10.2.2 s3c2440裸机下触摸屏控制器的接口编程
 - 10.3 嵌入式linux触摸屏驱动程序框架
 - 10.4 嵌入式linux触摸屏驱动的源码分析
 - 10.4.1 linux触摸设备驱动的处理流程
 - 10.4.2 触摸屏驱动模块的初始化函数s3c2410ts_init
 - 10.4.3 笔针按下中断服务处理程序stylus_updown
 - 10.5 嵌入式linux输入子系统的工作原理及实现机制
 - 10.5.1 linux输入子系统的主要数据结构与全局变量
 - 10.5.2 输入设备的注册流程
 - 10.5.3 事件处理器的注册流程
 - 10.5.4 输入事件的报告流程
 - 10.5.5 应用程序访问输入设备的流程
- 第11章 嵌入式linux mtd子系统与 flash驱动程序设计
 - 11.1 mtd子系统概述
 - 11.2 linux中nor flash驱动的源码分析
 - 11.3 mtd子系统的源码分析
 - 11.3.1 mtd子系统源码组织
 - 11.3.2 mtd子系统主要数据的结构分析
 - 11.4 nor flash芯片手册解读
- 第12章 嵌入式linux nand flash 驱动程序设计
 - 12.1 nand flash芯片硬件及接口介绍
 - 12.1.1 nand flash存储空间的组织
 - 12.1.2 nand flash的硬件接口及读写操作时序
 - 12.1.3 s3c2440对nand flash芯片的访问
 - 12.2 嵌入式linux下nand flash驱动分析

- 12.2.1 nand flash驱动源码组织
- 12.2.2 nand flash驱动架构
- 12.2.3 nand flash相关操作流程
- 12.2.4 s3c24xx_nand_probe函数分析
- 12.3 应用程序对nand flash设备的读/写操作
- 12.3.1 mtd字符设备写nand flash的操作分析
- 12.3.2 s3c2440_nand_hwcontrol函数
- 12.3.3 nand_command函数
- 第13章 嵌入式linux i2c总线驱动程序设计
- 13.1 i2c总线概述
- 13.2 s3c2440 i2c总线控制器的硬件工作原理
- 13.3 s3c2440 i2c控制器的硬件编程
- 13.3.1 初始化s3c2440 i2c主控制器
- 13.3.2 i2c总线写at24c02操作
- 13.3.3 i2c总线读at24c02操作
- 13.4 嵌入式linux i2c总线驱动架构
- 13.4.1 i2c体系架构的硬件实体
- 13.4.2 i2c驱动的软件实体
- 13.5 嵌入式linux i2c总线驱动源码的组织
- 13.6 嵌入式linux i2c总线控制器驱动的程序设计及源码分析
- 13.6.1 i2c总线控制器驱动的主要数据结构
- 13.6.2 写at24c02一个字节操作
- 13.6.3 i2c总线驱动框架
- 13.6.4 i2c总线控制器设备驱动探测函数probe的工作流程
- 13.6.5 i2c_add_adapter处理流程分析
- 13.6.6 定义和实现i2c适配器的底层操作接口algorithm
- 13.7 嵌入式linux i2c 设备驱动程序的设计及源码分析
- 13.7.1 i2c设备驱动程序框架
- 13.7.2 i2c_add_driver函数
- 13.7.3 at24c02b_probe函数
- 13.8 应用程序通过i2c设备驱动写at24c02一个字节的流程
- 第14章 嵌入式linux网卡驱动程序设计
- 14.1 概述
- 14.2 dm9000网络芯片与s3c2440的硬件原理图
- 14.3 dm9000a网卡芯片内部寄存器的访问
- 14.4 dm9000数据发送/接收的流程
- 14.5 嵌入式linux dm9000网卡驱动的框架及源码分析
- 14.5.1 dm9000设备
- 14.5.2 dm9000设备驱动
- 14.5.3 dm9000平台设备驱动的工作流程
- 14.5.4 应用层网络应用程序的操作
- 第15章 嵌入式linux usb设备驱动程序设计
- 15.1 usb通信系统概述
- 15.2 usb通信系统的拓扑结构图
- 15.3 usb通信的分时复用技术
- 15.4 usb通信系统的基本概念
- 15.5 usb通信的数据格式

- 15.5.1 域
- 15.5.2 包
- 15.5.3 事务
- 15.5.4 传输
- 15.5.5 usb标识域 (pid)
- 15.6 usb设备的枚举过程
- 15.7 usb设备端usb通信固件的程序设计
 - 15.7.1 usb芯片cy7c68013概述
 - 15.7.2 ez-usb固件程序的启动模式
 - 15.7.3 ez-usb芯片的中断系统
 - 15.7.4 usb固件程序的功能
 - 15.7.5 usb固件程序的框架及源码分析
- 15.8 嵌入式linux usb驱动程序框架
- 15.9 嵌入式linux usb主控制器驱动的源码分析
 - 15.9.1 s3c2440 usb主控制器平台设备驱动的源码分析
 - 15.9.2 usb主机驱动枚举usb设备的过程
- 15.10 嵌入式linux usb设备驱动的程序设计方法

章节摘录

版权页：插图：由于Linux内核触摸屏驱动基于输入子系统实现，对于触摸屏驱动开发人员来说，在大部分情况下只要自己编写与具体硬件相关的设备驱动源码即可，而事件核心层和事件处理层直接由内核提供，不需要再重新编写。

这里以S3C2440 CPU、四线电阻式触摸屏为例来分析Linux下触摸屏的设备驱动开发方法和思路，其他触摸屏的设备驱动开发与之类似。

10.4.1 Linux触摸设备驱动的处理流程 前面我们已经了解了触摸屏的工作原理，触摸屏的测量触摸位置信息的过程基本步骤如下：（1）初始化设置触摸屏控制器处于等待笔针按下状态，等待用户按下触摸笔。

（2）用户按下笔针后，触发笔针按下中断，在笔针按下中断服务处理程序中，开始测量X、Y位置坐标信息，X、Y坐标位置信息测量完成后，向输入子系统事件处理层报告位置信息及笔针状态，设置触摸屏控制器处于等待笔针抬起状态，等待用户抬起笔针。

（3）用户抬起笔针，触发笔针抬起中断，在笔针抬起中断服务程序中，再设置触摸屏控制器处于等待笔针按下状态，等待用户按下触摸笔，准备下一次触摸的测量。

Linux触摸屏驱动触摸点位置信息测量流程如下图所示。

了解了Linux下触摸屏驱动的总实现思路，下面分析Linux下触摸屏设备驱动的源码。

以S3C2440 ARM CPU为例，Linux触摸屏设备驱动的源码文件为s3c2410_ts.c。

Linux的触摸屏设备驱动是以模块方式存在的，编写一个Linux触摸屏设备驱动主要分为以下几个步骤：（1）初始化S3C2440的触摸屏控制器相关寄存器，设置ADC转换器的时钟频率的分频系数，A/D转换器的工作模式等。

（2）创建Linux内核中代表输入设备的全局结构体变量struct input_dev*input_dev，在Linux内核中，struct input_dev就代表一个设备驱动层的输入设备。

（3）设置设备驱动层输入设备struct input_dev所支持的事件类型及支持的按键编码。

（4）初始化输入设备struct input_dev的设备id，事件处理层通过设备的id来判断是否支持该输入设备。

（5）最后调用事件核心层提供的函数input_register_device来注册输入设备到核心层。

（6）触摸屏的设备驱动还需要注册两个中断以及完成两个中断处理函数，分别是触摸屏中断IRQ_TC和IRQ_ADC中断，IRQ_TC中断用来检测笔针按下和抬起信号；IRQ_ADC中断用来处理X、Y坐标位置信息的A/D转换及向事件处理层汇报输入事件。

了解了触摸屏驱动开发的基本步骤，接下来分析s3c2410_ts.C触摸屏设备的驱动源码。

触摸屏设备驱动s3c2410_ts.C是一个驱动模块，以模块的方式存在，将触摸屏驱动插入内核后，首先执行模块初始化函数s3c2410ts_init，我们就从s3c2410ts_init开始分析。

编辑推荐

《深入理解嵌入式Linux设备驱动程序》可作为高校嵌入式专业教材以及广大嵌入式爱好者的参考书。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>