

<<ARM9嵌入式系统设计>>

图书基本信息

书名：<<ARM9嵌入式系统设计>>

13位ISBN编号：9787512401945

10位ISBN编号：7512401949

出版时间：2010-9

出版时间：北京航空航天大学出版社

作者：徐英慧 等编著

页数：366

字数：602000

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>

<<ARM9嵌入式系统设计>>

前言

本书是2007年9月出版的《ARM9嵌入式系统设计——基于S3C2410与Linux》的第2版。

第2版对第1版的内容进行了全面的整理、优化和修改。

第1版中介绍的ARM开发环境主要是ADSI.2，第2版则主要介绍了目前流行的RealView MDK和IAREWARM。

第1版配套的开发平台是北京精仪达盛有限公司提供的EL-ARM—830，第2版则升级为EL-ARM-860，对相应的示例也进行了修整。

当前，嵌入式技术的应用越来越广泛，从航天科技到民用产品，嵌入式产品的身影无处不在，而这些嵌入式产品的核心——处理器决定了产品的市场和性能。

在32位嵌入式处理器市场中，ARM处理器占有很大的份额。

ARM不仅是一个公司、一种技术，也是一种经营理念，即由ARM公司提供核心技术，只出售芯片中的IP授权，采取了别具一格的“Chipless模式”（无芯片的芯片企业），不参与生产，而是由合作厂商去生产具体的芯片和产品。

现在由于存储空间等原因，在嵌入式芯片上编程有较大的困难，选取合适的平台就显得很重要。

Linux自出现以来，得到了迅速的发展。

Linux是开放源码的操作系统，吸引着全世界的程序员参与到发展和完善的工作中来，所以Linux保持了稳定而且卓越的性能。

Linux在服务器领域已经占有很大的份额，在图形界面方面也不输于windows。

由于源码可以修改、移植，Linux在嵌入式领域中的应用也越来越广。

选用Linux作为平台，可以根据具体需要自由地裁减源码，打造适合目标平台的环境，编写最有效率的应用程序。

<<ARM9嵌入式系统设计>>

内容概要

《ARM9嵌入式系统设计：基于S3C2410与Linux（第2版）

》针对在嵌入式市场上颇具竞争力的ARM9处理器——S3C2410和开放源码的Linux操作系统，讲述嵌入式系统的概念、软硬件的开发和调试手段、嵌入式Linux驱动程序和应用程序的开发以及图形用户界面Mini

GUI的移植和应用。

第2版将ADS1.2内容替换为RealView MDK和IAR

EWARM的内容，同时开发平台也升级为EL-ARM-860。

《ARM9嵌入式系统设计：基于S3C2410与Linux（第2版）

》的特点是集嵌入式系统开发的理论知识和实验教学于一体，并结合北京精仪达盛科技有限公司的开发板，给出了大量实例。

《ARM9嵌入式系统设计：基于S3C2410与Linux（第2版）

》可作为高等院校嵌入式系统课程的教材，也可作为对嵌入式系统开发感兴趣的读者的入门教材，同时还可以作为从事ARM嵌入式系统应用开发工程师的参考书。

<<ARM9嵌入式系统设计>>

书籍目录

第1章 嵌入式系统基础

1.1 嵌入式系统概念

1.1.1 嵌入式系统的定义

1.1.2 嵌入式系统的组成

1.1.3 嵌入式系统的特点

1.1.4 嵌入式系统的应用

1.1.5 实时系统

1.2 嵌入式处理器

1.2.1 嵌入式处理器分类

1.2.2 微控制器

1.2.3 嵌入式微处理器

1.2.4 DSP处理器

1.2.5 片上系统

1.2.6 典型的嵌入式处理器

1.3 嵌入式操作系统

1.3.1 操作系统的概念和分类

1.3.2 实时操作系统

1.3.3 常见的嵌入式操作系统

1.4 实时操作系统的内核

1.4.1 任务管理

1.4.2 任务间的通信和同步

1.4.3 存储器管理

1.4.4 定时器和中断管理

1.5 嵌入式技术发展现状及趋势

1.6 习题

第2章 嵌入式系统开发过程

2.1 嵌入式软件开发的特点

2.2 嵌入式软件的开发流程

2.3 嵌入式系统的调试

2.4 板级支持包

2.5 习题

第3章 ARM体系结构

3.1 ARM体系结构概述

3.1.1 ARM体系结构的特点

3.1.2 ARM处理器结构

3.1.3 ARM处理器内核

3.1.4 ARM处理器核

3.2 编程模型

3.2.1 数据类型

3.2.2 处理器模式

3.2.3 处理器工作状态

3.2.4 寄存器组织

3.2.5 异常

3.2.6 存储器和存储器映射I/O

3.3 ARM基本寻址方式

<<ARM9嵌入式系统设计>>

3.4 ARM指令集

3.4.1 条件执行

3.4.2 指令分类说明

3.4.3 Thumb指令集与ARM指令集的区别

3.4.4 Thumb指令分类说明

3.5 ARM9与ARM7的比较

3.6 ARM9TDMI内核

3.6.1 ARM9TDMI编程模型

3.6.2 ARM9TDMI存储器接口

3.7 ARM920T核

3.7.1 ARM920T编程模型

3.7.2 存储器管理单元 (MMU)

3.7.3 Cache写缓冲和物理地址TAGRAM

3.7.4 时钟模式

3.7.5 总线接口单元

3.8 习题

第4章 ARM系统硬件设计基础

4.1 ARM开发环境简介

4.1.1 RealView MDK开发工具简介

4.1.2 IAREWARM集成开发环境简介

4.2 基于ARM的汇编语言程序设计

4.2.1 ARM汇编器支持的伪指令

4.2.2 基于ARM的汇编语言语句格式

4.2.3 ARM汇编语言程序的基本结构

4.2.4 基于ARM的汇编语言程序举例

4.3 基于ARM的硬件启动程序设计

4.3.1 分配中断向量表

4.3.2 初始化存储系统

4.3.3 初始化堆栈

4.3.4 初始化有特殊要求的硬件模块

4.3.5 初始化应用程序执行环境

4.3.6 改变处理器模式

4.3.7 调用主应用程序

4.4 基于ARM的C语言与汇编语言混合编程

4.4.1 C语言与汇编语言混合编程应遵守的规则

4.4.2 汇编程序调用c程序的方法

4.4.3 C程序调用汇编程序的方法

4.4.4 C程序中内嵌汇编语句

4.4.5 基于ARM的C语言与汇编语言混合编程举例

4.5 印制电路板制作简介

4.5.1 印制电路板设计软件——Protel

4.5.2 单面板与多层板

4.5.3 印制电路板设计的注意事项

4.6 习题

第5章 基于S3C2410的系统硬件设计

5.1 S3C2410简介

5.1.1 S3C2410A的特点

<<ARM9嵌入式系统设计>>

- 5.1.2 存储器控制器
- 5.1.3 NAND Flash控制器
- 5.1.4 时钟和电源管理
- 5.2 I / O口
- 5.2.1 S3C2410A的I / O口工作原理
- 5.2.2 I / O口编程实例
- 5.3 中断
- 5.3.1 ARM的中断原理
- 5.3.2 S3C ; 2410A的中断控制器
- 5.3.3 中断编程实例
- 5.4 DMA
- 5.4.1 DMA工作原理
- 5.4.2 S3C2410A的DMA控制器
- 5.4.3 DMA编程实例
- 5.5 UART
- 5.5.1 UART的工作原理
- 5.5.2 S3C2410A的UART
- 5.5.3 UART编程实例
- 5.6 A / D接口
- 5.6.1 A / D接口原理
- 5.6.2 S3C ; 2410A的A / D转换器
- 5.6.3 A / D接口编程实例
- 5.7 键盘和LED控制
- 5.7.1 键盘和LED的接口原理
- 5.7.2 键盘和LED控制的编程实例
- 5.8 LCD
- 5.8.1 LCD显示原理
- 5.8.2 S3C2410A的LCD控制器
- 5.8.3 LCD显示的编程实例
- 5.9 触摸屏
- 5.9.1 触摸屏工作原理
- 5.9.2 S3C2410A的触摸屏接口
- 5.9.3 触摸屏编程实例
- 5.10 音频录放
- 5.10.1 音频录放的实现原理
- 5.10.2 S3C2410A的I2S总线接口
- 5.10.3 音频录放的编程实例
- 5.11 USB设备的数据收发
- 5.11.1 USB接口及编程简介
- 5.11.2 S3C2410A的USB设备控制器
- 5.11.3 USB设备收发数据编程实例
- 第6章 Linux操作系统基础
- 6.1 Linux操作系统概述
- 6.1.1 Linux操作系统的产生及
- 6.1.2 Linux操作系统的特点
- 6.2 Linux内核的结构
- 6.2.1 进程调度

<<ARM9嵌入式系统设计>>

- 6.2.2 内存管理
- 6.2.3 虚拟文件系统
- 6.2.4 进程间通信
- 6.2.5 网络接口
- 6.2.6 各个子系统之间的依赖关系
- 6.3 Linux设备管理
 - 6.3.1 设备文件
 - 6.3.2 设备驱动
 - 6.3.3 控制方式
- 6.4 Linux的使用
 - 6.4.1 Linux常用命令
 - 6.4.2 vi编辑器的使用
 - 6.4.3 make工具和gcc编译器
- 第7章 嵌入式Linux软件设计
 - 7.1 Bootloader引导程序
 - 7.1.1 Boot Loader的启动过程
 - 7.1.2 BootLoader——vivi
 - 7.2 Linux的移植
 - 7.2.1 Linux内核的目录结构
 - 7.2.2 Linux内核源码
 - 7.2.3 交叉编译环境的建立
 - 7.2.4 Linux内核文件的修改
 - 7.2.5 Linux内核及文件系统的编译
 - 7.2.6 Linux内核及文件系统的下载
 - 7.3 驱动程序开发
 - 7.3.1 驱动程序的开发步骤
 - 7.3.2 键盘驱动程序的开发
 - 7.3.3 LCD驱动程序的开发
 - 7.4 应用程序开发
 - 7.4.1 应用程序的开发步骤
 - 7.4.2 键盘应用程序的开发
 - 7.4.3 基本绘图应用程序的开发
 - 7.4.4 跑马灯应用程序的开发
- 第8章 图形用户接口Mini GUI
 - 8.1 Mini GUI简介
 - 8.1.1 Mini GUI的功能特色
 - 8.1.2 Mini GUI的技术优势
 - 8.2 Mini GUI在Linux下运行环境的建立
 - 8.2.1 Mini GUI在Linux下的运行环境
 - 8.2.2 安装资源文件
 - 8.2.3 配置安装Mini GUI库文件
 - 8.3 Mini GUI在QVFB上的仿真应用
 - 8.4 在Linux下S3C2410FrameBuffer的启动
 - 8.4.1 Frame Buffer简介
 - 8.4.2 Frame Buffer驱动的添加
 - 8.4.3 Frame Buffer设备文件的添加
 - 8.4.4 Frame Buffer测试程序的编写

<<ARM9嵌入式系统设计>>

8.5 Mini GUI在\$3C2410上的移植

8.5.1 安装Mini GUI资源文件

8.5.2 配置安装Mini GUI库文件

8.5.3 Mini GUI的移植步骤

8.6 Mini GUI输入引擎IAL的开发

8.6.1 IAL引擎简介

8.6.2 IAL引擎的开发

8.7 在C3C2410上运行简单的绘图程序

8.7.1 Mini GUI的基本绘图函数

8.7.2 绘图程序举例

附录AEL-ARM—860型嵌入式实验

开发系统简介

一、系统结构简介

二、可开设的实验项目

三、产品特点

附录B达盛科技Techv-DM355介绍

参考文献

<<ARM9嵌入式系统设计>>

章节摘录

插图：通用软件开发的设计常采用将系统划分为各个功能子模块，再进一步细分为函数，采用自顶向下的设计方法。

而嵌入式应用软件是通过并发的任务来运作的，设计时将系统划分为多个并发执行的任务，各个任务允许并发执行，通过相互间通信建立联系。

传统的设计方法不适应这种并发的设计模式，因而在嵌入式软件开发中引入了DARTS设计方法。

DARTS设计方法是结构化分析 / 结构化设计的扩展。

它给出划分任务的方法，并提供定义任务间接口的机制。

DARTS设计方法的设计步骤如下：（1）数据流分析在DARTS设计方法中，系统设计人员在系统需求的基础上，以数据流图作为分析工具，从系统的功能需求开始分析系统的数据流，确定系统的主要功能。

扩展系统的数据流图，分解系统到足够的深度，识别主要的子系统和各个子系统的主要成分。

（2）划分任务识别出系统的所有功能以及它们之间的数据流关系，并得到完整的数据流图后，下一步是识别出可并行的功能。

系统设计人员把可并行的、相对独立的功能单元抽象成一个系统任务。

DARTS设计方法提供了在数据流图上确定并发任务的方法。

实时软件系统中并行任务的分解主要考虑系统内功能的异步性。

根据数据流图中的变换，分析出哪些变换是可以并行的，哪些变换是顺序执行的。

系统设计人员可以考虑一个变换对应一个任务，或者一个任务包括多个变换

<<ARM9嵌入式系统设计>>

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>