

<<ARM处理器裸机开发实战>>

图书基本信息

书名：<<ARM处理器裸机开发实战>>

13位ISBN编号：9787121153037

10位ISBN编号：7121153033

出版时间：2012-1

出版时间：电子工业出版社

作者：王小强 主编

页数：369

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>

<<ARM处理器裸机开发实战>>

内容概要

本书面向立志于进行ARM嵌入式开发的初学者以及从单片机向ARM处理器转型的工程师，按照理论与实践相结合的思想，介绍了ARM嵌入式开发过程中的基础理论，并给出了具体的实例。全书共分为4篇，包括ARM汇编语言、ARMC语言、ARM处理器各功能模块开发等内容。

本书针对ARM处理器裸机开发过程中的重点、难点问题，既有基础知识的讲述，又有相关配套实验，使读者能容易、快速、全面地掌握ARM处理器裸机开发。

<<ARM处理器裸机开发实战>>

书籍目录

第1篇 基础篇

第1章 ARM处理器简介

1.1 处理器工作模式

1.2 寄存器介绍

1.2.1 堆栈指针寄存器R13和链接寄存器R14

1.2.2 程序计数器R15

1.2.3 程序状态寄存器

1.3 工作状态

1.4 数据长度

1.5 存储系统

1.5.1 ARM地址空间

1.5.2 ARM存储器的格式

1.6 天嵌TQ2440开发板硬件资源概述

1.7 本章小结

1.8 扩展阅读之CISC处理器和RISC处理器简介

第2章 ADS集成开发环境及程序下载具体流程

2.1 ADS 1.2集成开发环境简介

2.1.1 CodeWarrior for ARM

2.1.2 AXD调试器的启动

2.2 工程的编辑与修改

2.2.1 建立一个新工程

2.2.2 建立一个源文件

2.2.3 添加源文件到工程

2.2.4 编译与链接工程

2.2.5 打开已有的工程

2.3 工程的调试

2.3.1 装载映像文件

2.3.2 调试工具条的使用

2.4 H-JTAG的安装与调试

2.4.1 H-JTAG的安装

2.4.2 H-JTAG的设置

2.5 使用U-Boot下载裸机程序

2.6 本章小结

第3章 ARM指令集及汇编语言基础

3.1 ARM指令集介绍

3.1.1 ARM指令集

3.1.2 ARM寻址方式

3.1.3 ARM伪操作和伪指令介绍

3.2 ARM汇编基础知识

3.3 ARM汇编程序的基本结构

3.3.1 编写汇编程序基本的格式规范

3.3.2 程序入口和程序结束

3.3.3 段

3.3.4 标号(标志符)

3.3.5 外部标号

<<ARM处理器裸机开发实战>>

- 3.3.6 文件包含
- 3.4 用AXD调试ARM汇编程序实验
 - 3.4.1 建立工程并添加源文件
 - 3.4.2 工程的设置
 - 3.4.3 编译源文件
 - 3.4.4 启动AXD调试器
 - 3.4.5 手把手调试汇编程序
- 3.5 常用汇编语言程序子模块实例分析
 - 3.5.1 特殊功能寄存器的访问
 - 3.5.2 内存数据复制
 - 3.5.3 批量加载与存储
 - 3.5.4 堆栈操作
 - 3.5.5 实现查表功能
- 3.6 本章小结
- 3.7 扩展阅读之内存和I/O地址、前序寻址和后序寻址
- 第4章 ARM C语言基础
 - 4.1 数据类型基础
 - 4.1.1 用typedef和#define定义类型
 - 4.1.2 用signed和unsigned修饰数据类型
 - 4.1.3 volatile和强制类型转换
 - 4.2 深入理解位运算符和位运算
 - 4.2.1 按位与运算符 (&)
 - 4.2.2 按位或运算符 (|)
 - 4.2.3 按位取反运算符 (~)
 - 4.2.4 左移和右移运算符 (< <)、 (> >)
 - 4.2.5 位运算应用实例分析
 - 4.3 控制结构
 - 4.3.1 选择结构
 - 4.3.2 循环结构
 - 4.4 防止文件重复包含技巧
 - 4.5 ARM编译器对C语言的扩展
 - 4.5.1 __irq声明中断处理函数
 - 4.5.2 __swi声明软中断
 - 4.5.3 __asm内嵌汇编
 - 4.5.4 __inline定义内联函数
 - 4.6 本章小结
 - 4.7 扩展阅读之高速缓存基础知识
- 第5章 ARM汇编语言和C语言混合编程基础
 - 5.1 一个混合编程实例的实现
 - 5.2 APCS规则概述
 - 5.2.1 寄存器的使用
 - 5.2.2 参数传递
 - 5.2.3 函数的返回值
 - 5.3 本章小结
- 第6章 GPIO编程实验
 - 6.1 GPIO概述
 - 6.1.1 GPIO引脚介绍

<<ARM处理器裸机开发实战>>

6.1.2 GPIO特性分析

6.1.3 GPIO相关寄存器

6.1.4 GPIO应用实例

6.2 基础实验：第一个裸机程序——流水灯

6.2.1 硬件电路分析

6.2.2 建立工程并添加启动代码

6.2.3 添加源文件

6.2.4 编辑源文件

6.2.5 工程设置、编译、链接

6.2.6 下载程序到开发板运行

6.2.7 由点亮LED引发的思考

6.2.8 再议点亮LED实验

6.2.9 将点亮一个LED扩展到流水灯

6.3 GPIO扩展实验

6.3.1 按键实验

6.3.2 蜂鸣器实验

6.4 本章小结

6.5 扩展阅读之模块化编程、NAND FLASH和 NOR FLASH概述

第2篇 提高篇

第7章 启动代码分析

7.1 从开发板硬件讲起

7.1.1 TQ2440核心板芯片功能介绍

7.1.2 从NAND FLASH和NOR FLASH启动流程分析

7.2 启动代码详解

7.3 启动代码主要功能模块分析

7.3.1 建立中断向量表

7.3.2 初始化各个模式的堆栈

7.3.3 初始化系统硬件

7.3.4 初始化应用程序的执行环境并跳转到主程序执行

7.3.5 跳转到C语言主程序执行

7.4 本章小结

7.5 本章附录——完整版启动代码

第8章 系统时钟和定时器

8.1 S3C2440时钟系统概述

8.1.1 系统时钟初始化

8.1.2 FCLK、HCLK和PCLK与Fin的关系

8.2 定时器原理与应用

8.2.1 定时器原理

8.2.2 定时器相关的寄存器

8.2.3 定时器基础实验代码详解及测试

8.2.4 定时器扩展实验之PWM实验

8.3 本章小结

第9章 存储器控制器

9.1 S3C2440地址空间

9.2 操作实例：SDRAM实例分析

9.2.1 SDRAM工作原理

9.2.2 SDRAM接口电路设计

<<ARM处理器裸机开发实战>>

- 9.2.3 SDRAM初始化过程详解
- 9.2.4 回顾启动代码中的SDRAM初始化
- 9.3 本章小结
- 第10章 通用异步收发器 (UART)
 - 10.1 UART概述
 - 10.2 S3C2440处理器UART工作原理
 - 10.3 引脚描述及相关寄存器
 - 10.4 UART基础实验
 - 10.4.1 硬件电路分析
 - 10.4.2 程序设计及代码详解
 - 10.4.3 实例测试
 - 10.4.4 UART基础实验分析
 - 10.5 UART高级实验——可变参数函数在UART中的应用
 - 10.5.1 程序设计及代码详解
 - 10.5.2 实例测试
 - 10.6 本章小结
- 第11章 中断控制系统
 - 11.1 S3C2440中断系统概述
 - 11.1.1 深入理解CPU的工作模式
 - 11.1.2 中断控制器
 - 11.2 外部中断实验
 - 11.2.1 硬件电路分析
 - 11.2.2 程序分析
 - 11.2.3 中断执行流程详解
 - 11.2.4 中断处理流程引发的思考
 - 11.2.5 实例测试
 - 11.2.6 为什么进入不了中断
 - 11.3 定时器中断实验
 - 11.3.1 程序代码分析
 - 11.3.2 实例测试
 - 11.4 串口中断原理及实验
 - 11.4.1 如何正确使用中断
 - 11.4.2 程序代码分析
 - 11.4.3 实例测试
 - 11.5 ARM中断之高级应用：软中断原理及实验
 - 11.5.1 程序代码分析
 - 11.5.2 实例测试
 - 11.5.3 软中断所用到的启动代码
 - 11.6 本章小结
- 第12章 NAND FLASH原理与实验
 - 12.1 FLASH概述
 - 12.1.1 NAND FLASH的基本结构
 - 12.1.2 NAND FLASH接口电路
 - 12.1.3 如何访问NAND FLASH
 - 12.1.4 S3C2440 NAND FLASH控制器
 - 12.1.5 使用宏代替简单的函数
 - 12.2 NAND FLASH基础实验

<<ARM处理器裸机开发实战>>

- 12.2.1 NAND FLASH基本操作函数分析
- 12.2.2 NAND FLASH基础实验之页读写
- 12.2.3 页读写实例测试
- 12.2.4 NAND FLASH基础实验之读ID
- 12.2.5 读ID实例测试
- 12.3 NAND FLASH高级实验
 - 12.3.1 随机读、写实验代码详解
 - 12.3.2 随机读、写实例测试
- 12.4 回顾启动代码中的NAND FLASH读取函数
- 12.5 本章小结
- 第13章 LCD控制器原理与实验
 - 13.1 LCD和LCD控制器工作原理
 - 13.1.1 LCD概述
 - 13.1.2 LCD接口信号
 - 13.1.3 LCD显示原理
 - 13.1.4 LCD操作时序详解
 - 13.1.5 S3C2440 LCD控制器
 - 13.1.6 LCD 控制寄存器初始化
 - 13.2 LCD基础实验
 - 13.2.1 硬件电路分析
 - 13.2.2 程序代码分析
 - 13.2.3 实例测试
 - 13.3 LCD基础实验之单像素显示
 - 13.3.1 程序代码分析
 - 13.3.2 实例测试
 - 13.4 LCD基础实验之图片显示
 - 13.4.1 如何将图片转换为C语言数组
 - 13.4.2 程序代码分析
 - 13.4.3 实例测试
 - 13.5 LCD高级实验之汉字显示
 - 13.5.1 两种常见的汉字编码
 - 13.5.2 LCD汉字显示原理
 - 13.5.3 程序代码分析
 - 13.5.4 实例测试
 - 13.5.5 LCD显示高级技巧——可变参函数Lcd_Printf的实现
 - 13.5.6 可变参函数Lcd_Printf测试
 - 13.5.7 汉字区位码的思考
 - 13.5.8 实例测试
 - 13.6 本章小结
- 第14章 ADC原理与实验
 - 14.1 ADC原理
 - 14.1.1 ADC相关寄存器
 - 14.1.2 ADC初始化
 - 14.2 ADC实验
 - 14.2.1 ADC实验代码详解
 - 14.2.2 ADC实验测试
 - 14.3 本章小结

<<ARM处理器裸机开发实战>>

第3篇 典型项目分析

第15章 综合实战

15.1 实战1：数据采集系统实现

15.1.1 功能描述

15.1.2 模块划分

15.1.3 代码实现

15.1.4 实例测试

15.1.5 实验总结

15.2 实战2：串口控制实验

15.2.1 功能描述

15.2.2 模块划分

15.2.3 代码实现

15.2.4 实例测试

15.2.5 实验总结

15.3 实战3：制作电子相册

15.3.1 功能描述

15.3.2 模块划分

15.3.3 代码实现

15.3.4 实例测试

15.3.5 实验总结

第4篇 理论知识扩展

第16章 嵌入式系统电源设计和Linux内核开发基础

16.1 直流稳压电源分类

16.1.1 普通线性稳压器工作原理

16.1.2 低压差线性稳压器工作原理

16.1.3 电容式开关电源的工作原理

16.1.4 电感式开关电源的工作原理

16.1.5 嵌入式系统设计中的电源芯片选型

16.1.6 设计实例分析

16.2 Linux内核基础实验

16.2.1 实验一：修改调度算法实验

16.2.2 实验二：添加内核模块实验

参考文献

<<ARM处理器裸机开发实战>>

章节摘录

版权页：插图：最初的ARM处理器由英国剑桥的Acorn计算机公司（是ARM公司的前身）设计。

ARM公司成立于1990年，该公司是知识产权（IP）提供商（不生产芯片）。

目前，ARM架构处理器已在高性能、低功耗、低成本的嵌入式应用领域中占据了领先地位。

ARM公司作为嵌入式RISC处理器的知识产权IP供应商，本身并不直接从事芯片生产，而是将设计许可授权给合作公司，合作公司添加自己的外设，进而生产各具特色的SOC芯片，利用这种合伙关系，ARM很快成为许多全球性RISC标准的缔造者。

目前，全世界有几十家大的半导体公司都使用ARM公司的授权，其中包括Intel、IBM、Samsung、LG半导体、NEC、SONY、PHILIP等公司。

因此，采用ARM处理器进行嵌入式系统开发时，开发者可以获得更多的第三方工具和技术支持，进而从一定程度上降低整个系统的研发成本，缩短研发周期，从而使产品更具市场竞争力。

ARM体系结构基于精简指令集计算机（RISC）原理。

RISC的相关译码机制比复杂指令集计算机（CISC）的设计更简单，从而有更高的指令吞吐率、出色的实时中断响应。

ARM处理器的特性：

<<ARM处理器裸机开发实战>>

编辑推荐

《ARM处理器裸机开发实战:机制而非策略》：写作特点 理论和实践相结合，夯实理论基础，强化实践环节、 模块化设计与系统设计相结合、 立足于ARM处理器，并给出了设计思想与方案、 用朴实的语言描述看似深奥的理论。

适用对象 高等院校电子、通信、自动控制等专业学生、 从单片机开发向ARM嵌入式开发转型的工程师、 从事ARM嵌入式开发的相关技术人员。

<<ARM处理器裸机开发实战>>

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>