

<<物联网与嵌入式系统开发>>

图书基本信息

书名：<<物联网与嵌入式系统开发>>

13位ISBN编号：9787121179976

10位ISBN编号：7121179970

出版时间：2012-9

出版时间：电子工业出版社

作者：刘连浩

页数：362

字数：520000

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>

<<物联网与嵌入式系统开发>>

内容概要

刘连浩编著的《物联网与嵌入式系统开发》是依托中南大学国家级特色专业（物联网工程）的建设，结合国内物联网工程专业的教学情况编写的。

本书主要介绍物联网应用中的嵌入式系统的开发，以S3C2440为硬件，以嵌入式Linux为软件，详细介绍了物联网与嵌入式系统的关系、ARM

Cortex的体系结构、ARM的指令和程序设计、S3C2440的硬件结构与编程、嵌入式Linux的开发和移植等内容，最后给出了在物联网应用中常用的嵌入式系统开发实例。

《物联网与嵌入式系统开发》可作为普通高等学校物联网工程及其相关专业的教材，也可供从事物联网及其相关专业的人士阅读。

<<物联网与嵌入式系统开发>>

书籍目录

第1章 嵌入式系统概述

- 1.1 嵌入式系统简介
- 1.2 嵌入式系统的组成
- 1.3 嵌入式处理器简介
- 1.4 嵌入式系统应用
- 1.5 嵌入式系统发展趋势
- 1.6 物联网与嵌入式系统
- 1.7 嵌入式开发工具
 - 1.7.1 软件开发工具
 - 1.7.2 硬件开发工具

思考与习题

第2章 ARM体系结构

- 2.1 ARM处理器基础
 - 2.1.1 ARM简介
 - 2.1.2 冯诺依曼结构、哈佛结构与改进的哈佛结构
 - 2.1.3 CISC与RISC
 - 2.1.4 流水线
 - 2.1.5 ARM处理器的命名方式
- 2.2 ARM处理器系列
 - 2.2.1 ARM处理器架构
 - 2.2.2 常用ARM处理器系列简介
- 2.3 ARM体系结构
 - 2.3.1 ARM的基本数据类型
 - 2.3.2 ARM处理器的运行模式
 - 2.3.3 ARM处理器的运行状态
 - 2.3.4 存储器的大/小端
 - 2.3.5 ARM的寄存器组织
 - 2.3.6 异常处理
- 2.4 ARM存储器结构
 - 2.4.1 ARM存储器接口
 - 2.4.2 ARM Cache结构
 - 2.4.3 ARM存储器管理单元MMU
 - 2.4.4 地址变换后备缓冲器
 - 2.4.5 ARM的MMU结构
- 2.5 ARM的其他部件
 - 2.5.1 存储器映像I/O
 - 2.5.2 直接存储器存取
 - 2.5.3 中断IRQ和快速中断FIQ
 - 2.5.4 ARM协处理器接口
 - 2.5.5 ARM AMBA接口

思考与习题

第3章 ARM指令集与程序设计

- 3.1 ARM指令系统特点
 - 3.1.1 数据类型
 - 3.1.2 ARM指令特点

<<物联网与嵌入式系统开发>>

3.2 ARM指令系统

3.2.1 ARM指令的寻址方式

3.2.2 ARM指令的条件执行

3.2.3 Load/Store类指令

3.2.4 ARM数据处理类指令

3.2.5 ARM转移类指令

3.3 Thumb指令

3.3.1 Thumb指令概述

3.3.2 Thumb存储器访问指令

3.3.3 Thumb数据处理指令

3.3.4 分支指令

3.3.5 中断和断点指令

3.4 ARM伪操作与伪指令

3.4.1 符号定义与变量赋值伪操作

3.4.2 数据定义伪操作

3.4.3 汇编控制伪操作

3.4.4 信息报告伪操作

3.4.5 指令集选择伪操作

3.4.6 杂项伪操作

3.4.7 ADR、ADRL、LDR伪指令

3.4.8 NOP伪指令

3.5 ARM汇编程序设计

3.5.1 汇编语言的基本格式

3.5.2 汇编语言的子程序调用

3.6 ARM C语言基础及混合编程

3.6.1 ATPCS概述

3.6.2 C语言与汇编语言混合编程

3.7 超级终端的使用

3.8 ADS集成开发环境的使用

3.8.1 安装与卸载ADS

3.8.2 使用ADS创建工程

3.8.3 使用AXD进行仿真

思考与习题

第4章 基于S3C2440的硬件结构与接口编程

4.1 S3C2440简介

4.1.1 S3C2440A片上主要功能

4.1.2 S3C2440A特点

4.2 S3C2440A的存储器接口

4.2.1 S3C2440A存储器控制器

4.2.2 NAND Flash控制器

4.2.3 SDRAM原理

4.3 时钟和电源管理

4.4 DMA

4.4.1 DMA请求源

4.4.2 DMA工作过程

4.4.3 基本DMA时序

4.4.4 DMA传输大小

<<物联网与嵌入式系统开发>>

4.4.5 DMA专用寄存器

4.5 I/O端口

4.6 定时器

4.7 ARM中断

4.8 LCD

4.8.1 LCD原理

4.8.2 ARM LCD驱动控制

4.8.3 LCD控制器特殊寄存器

4.9 A/D转换器和触摸屏

4.9.1 A/D转换器简介

4.9.2 触摸屏的工作原理及种类

4.9.3 A/D转换器和触摸屏接口

4.10 异步串口通信

4.10.1 串行通信概述

4.10.2 S3C2440A异步串行口简介

4.11 SPI、IIC、IIS、SD卡总线接口

4.11.1 SPI总线接口

4.11.2 IIC总线接口

4.11.3 IIS总线接口

4.11.4 SD卡

4.12 USB Host/Device控制器

思考与习题

第5章 嵌入式操作系统

5.1 操作系统简介

5.1.1 操作系统

5.1.2 嵌入式操作系统

5.2 C/OS-II简介

5.3 VxWorks简介

5.4 WinCE简介

5.5 Linux简介

5.5.1 Linux简介

5.5.2 Linux特点

5.5.3 嵌入式Linux

5.5.4 Linux内核版本与发行版

5.5.5 Linux进程管理

5.5.6 存储管理

5.5.7 文件系统

5.5.8 设备管理

5.5.9 Linux内核模块

5.5.10 Linux配置文件

5.5.11 Linux启动流程简介

5.6 其他嵌入式操作系统

5.6.1 其他国外常用嵌入式操作系统

5.6.2 其他国内常用的嵌入式操作系统

思考与习题

第6章 嵌入式Linux应用开发

6.1 Linux安装

<<物联网与嵌入式系统开发>>

- 6.1.1 安装vmware
- 6.1.2 创建虚拟机
- 6.1.3 安装fedora14
- 6.1.4 安装VMware Tools
- 6.2 交叉编译环境的建立
 - 6.2.1 交叉编译
 - 6.2.2 交叉编译器的安装
- 6.3 Linux常用命令
 - 6.3.1 系统管理命令
 - 6.3.2 文件管理命令
 - 6.3.3 文件查看命令
 - 6.3.4 进程管理命令
 - 6.3.5 网络管理命令
 - 6.3.6 用户管理命令
- 6.4 minicom的使用
 - 6.4.1 minicom的安装
 - 6.4.2 minicom的设置
 - 6.4.3 minicom的使用
- 6.5 vi程序的使用
 - 6.5.1 启动vi
 - 6.5.2 常用vi命令
- 6.6 gcc程序的使用
 - 6.6.1 gcc简介
 - 6.6.2 gcc编译程序的流程
 - 6.6.3 gcc选项
- 6.7 gdb程序的使用
 - 6.7.1 gdb简介
 - 6.7.2 gdb的调试程序流程
 - 6.7.3 gdb命令
- 6.8 makefile简介
 - 6.8.1 makefile简介
 - 6.8.2 makefile的规则
 - 6.8.3 makefile文件的命名
 - 6.8.4 makefile的组成
 - 6.8.5 简单的makefile示例
 - 6.8.6 使用变量
 - 6.8.7 使用隐式规则
 - 6.8.8 伪目标
- 6.9 Linux应用程序编程
 - 6.9.1 文件编程
 - 6.9.2 进程创建
 - 6.9.3 进程间通信之管道
 - 6.9.4 进程间通信之有名管道
 - 6.9.5 进程间通信之信号
 - 6.9.6 进程间通信之信号量
 - 6.9.7 进程间通信之消息队列
 - 6.9.8 进程间通信之共享内存

<<物联网与嵌入式系统开发>>

6.9.9 原子操作

6.9.10 线程

6.9.11 网络编程基础

6.9.12 Linux中断

思考与习题

第7章 嵌入式Linux操作系统的移植

7.1 Bootloader

7.1.1 Bootloader简介

7.1.2 常见Bootloader

7.2 Linux的移植

7.2.1 安装前的准备工作

7.2.2 Linux的移植步骤

7.3 根文件系统的制作

思考与习题

第8章 物联网中的常用嵌入式系统

8.1 TinyOS简介

8.1.1 TinyOS概述

8.1.2 TinyOS的特点

8.1.3 TinyOS开发平台

8.1.4 TinyOS开发案例

8.1.5 TinyOS的基本概念

8.2 安装TinyOS

8.3 nesC简介

8.3.1 nesC概述

8.3.2 nesC基本概念

8.3.3 一个简单的nesC编程示例

8.3.4 TOSSIM仿真

8.4 TinyOS内部机制简介

8.4.1 TinyOS程序运行机制分析

8.4.2 TinyOS的调度机制

8.4.3 TinyOS的通信模型

8.4.4 TinyOS的能量管理

思考与习题

参考文献

章节摘录

版权页：插图：3.6.1 APCS概述 APCS (ARM—Thumb Produce Call Standard) 是ARM程序和Thumb程序中子程序调用的基本规则，目的是为了使单独编译的C语言程序和汇编程序之间能够相互调用。

这些基本规则包括子程序调用过程中寄存器的使用规则、数据栈的使用规则和参数的传递规则。

1.寄存器的使用规则（见表34）子程序间通过寄存器R0~R3来传递参数。

被调用的子程序在返回前无须恢复寄存器R0~R3的内容。

在子程序中，使用寄存器R4~R11来保存局部变量。

这时，寄存器R4~R11可以记为v1~v8。

如果在子程序中使用了寄存器v1~v8中的某些寄存器，则子程序进入时必须保存这些寄存器的值，在返回前必须恢复这些寄存器的值。

在Thumb程序中，通常只能使用寄存器R4~R7来保存局部变量。

另外R9、R10和R11还有一个特殊作用，分别记为静态基址寄存器SB、数据栈限制指针SL和帧指针FP。

寄存器R12用做过程调用中间临时寄存器IP。

寄存器R13用做堆栈指针SP。

在子程序中寄存器R13不能用做其他用途。

寄存器SP在进入子程序时的值和退出子程序的值必须相等。

寄存器R14称为链接寄存器LR，它用于保存子程序的返回地址。

如果在子程序中保存了返回地址，寄存器R14则可以用做其他用途。

寄存器R15为程序计数器PC，不能用做其他用途。

只有寄存器R0~R7、SP、LR和PC可以在Thumb状态下使用，其中R7常常作为Thumb状态的工作寄存器，记为WR。

2.数据栈使用规则 栈指针通常可以指向不同的位置。

当栈指针指向栈顶元素（即最后一个入栈的数据元素）时，称为FULL栈。

当栈指针指向与栈顶元素相邻的一个元素时，称为Empty栈。

数据栈的增长方向也可以不同。

当数据栈向内存减小的地址方向增长时，称为Descending栈，当数据栈向着内存地址增加的方向增长时，称为Ascending栈。

综合这两种特点可以由以下4种数据栈：FD、ED、FA、EA。

ATPCS规定数据栈为FD类型，并对数据栈的操作是8字节对齐的，下面是一个数据栈的示例及相关的名词。

数据栈栈指针（Stack Pointer）：指向最后一个写入栈的数据的内存地址。

数据栈的基地址（Stack Base）：是指数据栈的最高地址。

由于ATPCS中的数据栈是FD类型的，实际上数据栈中最早入栈数据占据的内存单元是基地址的下一个内存单元。

数据栈界限（Stack Limit）：是指数据栈中可以使用的最低的内存单元地址。

已占用的数据栈（Used Stack）：是指数据栈的基地址和数据栈栈指针之间的区域。

其中包括数据栈栈指针对应的内存单元。

数据栈中的数据帧（Stack Frames）：是指在数据栈中，为子程序分配的用来保存寄存器和局部变量的区域。

异常中断的处理程序可以使用被中断程序的数据栈，这时用户要保证中断的程序数据栈足够大。

使用ADS编译器产生的目标代码中包含了DRFAT2格式的数据帧。

在调试过程中，调试器可以使用这些数据帧来查看数据栈中的相关信息。

而对于汇编语言来说，用户必须使用FRAME伪操作来描述数据栈中的数据帧。

ARM汇编器根据这些伪操作在目标文件中产生相应的DRFAT2格式的数据帧。

<<物联网与嵌入式系统开发>>

编辑推荐

《国家级特色专业(物联网工程)规划教材:物联网与嵌入式系统开发》可作为普通高等学校物联网工程及其相关专业的教材,也可供从事物联网及其相关专业的人士阅读。

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>