

<<Cortex-M3 +  $\mu$  C/OS-II>>

图书基本信息

书名：<<Cortex-M3 +  $\mu$  C/OS-II嵌入式系统开发入门与应用>>

13位ISBN编号：9787115231055

10位ISBN编号：7115231052

出版时间：2010-8

出版时间：人民邮电

作者：陈瑶//李佳//宋宝华

页数：303

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>

## 前言

Cortex-M3是ARM公司推出的最新的针对微控制器应用的内核，它提供了业界领先的高性能和低成本的解决方案，将成为MCU（Micro Controller Unit，微控制单元）应用的热点和主流。

Cortex-M3是一个32位的核，在传统的单片机领域中，有一些不同于通用32位CPU应用的要求，谭军博士曾说，在工控领域，用户要求具有更快的中断速度，而Cortex-M3采用了Tail-Chaining中断技术，完全基于硬件进行中断处理，最多可减少12个时钟周期，在实际应用中可减少70%的中断。

完全符合用户的需求。

Cortex-M3采用了新型的单线调试（Single Wire）技术，专门拿出一个引脚来做调试，从而节约了大笔调试工具的费用。

同时，Cortex-M3中还集成了大部分存储器、控制器，这样工程师可以直接在MCU外连接Flash，降低了设计和应用的难度。

## <<Cortex-M3 + $\mu$ C/OS-II>>

### 内容概要

本书首先讲解了Cortex-M3处理器的架构及RealView MDK开发平台的相关知识，然后对 $\mu$ C/OS-的内核移植、驱动开发、应用程序开发进行了介绍，最后综合运用所学知识介绍了一个 $\mu$ C/OS- 的程序设计实例。

本书内容由浅入深，为加深理解，列举了很多程序设计的相关实例。

本书适合嵌入式初学者、嵌入式开发人员以及高校相关专业师生阅读。

## 作者简介

李佳，高级软件工程师，曾任职于微软中国研究院。  
现任职于东软集团，担任项目经理一职，负责客户关系管理系统在电信行业中的应用。  
从事软件开发近10年时间，曾主持开发过某省教育厅学籍学历查询系统。  
清华大学土木水利学院实验室管理信息系统。  
网上购物平台等各种大型系统。  
熟悉各种开发语言和工具，尤其擅长C#。  
对软件开发有深刻的认识，具有丰富的应用型软件开发经验。

### 宋宝华

华清远见金牌讲师，著有《Linux设备驱动开发详解》，译有《Essential Linux Device Drivers》，曾为思科、阿尔卡特、朗讯、四方、意法半导体等知名企业提供嵌入式Linux企业培训和技术服务，受到企业广泛好评。  
活跃于Linux社区，已给Linux内核贡献了逾2万行代码。

### 陈瑶

华清远见特邀顾问，任教于北京工业大学计算机学院，计算机软件与理论系。  
长期以来从事软件工程和嵌入式开发方面的教学与科研工作。  
曾于2004年作为访问学者在美国University of Massachusetts进修。  
参加过多项国家自然科学基金、北京市自然科学基金项目，发表多篇论文。

<<Cortex-M3 +  $\mu$ C/OS-II>>

## 书籍目录

第1部分 第1章 走进ARM微处理器	1.1 ARM体系结构的源头——ARM公司	1.2 ARM
处理器的发展历程	1.3 ARM体系结构	1.3.1 ARM Cortex处理器技术特点
ARM Cortex系列处理器比较	1.4 ARM v7新增指令	1.4.1 位操作指令
比特反转指令	1.4.3 16-bit常数操作	1.4.4 调转表指令
1.4.6 检查是否为零并跳转指令(CBZ)	第2章 Cortex-M3处理器体系结构	2.1 Cortex-M3
综述	2.2 Cortex-M3编程模式	2.2.1 Cortex-M3的工作模式和工作状态
特权访问和用户访问	2.2.3 Cortex-M3的寄存器组织	2.2.4 Cortex-M3的数据类型
2.2.5 Cortex-M3的存储器格式	2.3 Cortex-M3的存储器管理	2.3.1 存储器映射
2.3.2 Bit-banding机制	2.4 Cortex-M3的异常处理	2.4.1 异常优先级
2.4.2 异常处理的堆栈使用	2.4.3 Cortex-M3特有的异常处理机制	2.4.4 异常退出
2.4.5 复位异常	2.4.6 中止(Abort)异常	2.5 Cortex-M3的电源管理
2.5.1 SLEEPING	2.5.2 SLEEPDEEP	2.6 嵌套向量中断控制器NVIC
器保护单位MPU	2.8 Cortex-M3开发平台——STM32V100评估板介绍	2.7 存储
境RealView MDK平台搭建	第4章 基于RealView开发环境的嵌入式软件开发	第3章 ARM开发环
处理器内部资源C编程与实例	第5章 STM32F103	第4章 基于RealView开发环境的嵌入式软件开发
第2部分	第6章 $\mu$ C/OS- 操作系统基础及其移植开发初步	第5章 STM32F103
$\mu$ C/OS- 的内核机制	第7章	第6章 $\mu$ C/OS- 操作系统基础及其移植开发初步
第8章 基于 $\mu$ C/OS- 的程序设计实例	参考文献	第7章

## 章节摘录

插图：实时性：响应外部事件的时间必须在限定的时间范围内，在某些情况下还需要是确定的、可重复实现的，不管当时系统内部状态如何，都必须是可预测的。

抢占式调度：为确保响应时间，实时操作系统必须允许高优先级的任务一旦进入就绪状态，就可以马上抢占正在运行的低优先级任务的执行权。

具有异步响应能力：异步事件是指无一定时序关系、随机发生的事件。

如实时控制设备出现异常等突发事件，都属于随机事件。

实际环境中，嵌入式实时系统需要处理多个外部事件，这些事件往往同时出现，而且发生的时刻也是随机的。

实时操作系统应有能力对这类同时发生的外部事件进行有效的处理。

内存锁定：必须具有将程序部分代码锁定在内存的能力，将频繁访问的数据锁定在内存，减少了为获得该数据而访问磁盘的时间，从而保证了快速的响应时间。

具有优先级调度机制：实时操作系统必须允许用户定义中断和任务的优先级，并具有相应的优先级调度机制。

同步 / 互斥机制：提供对共享数据的同步和互斥手段。

实时操作系统能对外部事件和信号在限定的时间范围内作出响应，它所强调的是实时性、可靠性和灵活性。

实时操作系统一般与实时应用软件相结合成为有机整体：用实时操作系统来管理和调度实时应用软件的各项任务，为应用软件提供良好的运行和开发环境。

一般来说，实时操作系统以库的形式提供系统调用来实现对外层实时应用程序的支持；而应用程序通过链接实时操作系统的库来实现实时任务调度。

## <<Cortex-M3 + $\mu$ C/OS-II>>

### 编辑推荐

《Cortex-M3 +  $\mu$  C/OS-II嵌入式系统开发入门与应用》：详细介绍Cortex—M3的内核结构、编程模式及其开发工具从Cortex-M3 +  $\mu$  C/OS-II上的移植入手，介绍在Cortex—M3平台上用 $\mu$  C/OS-II操作系统开发的方法通过一个Cortex-M3 +  $\mu$  C/OS-II完整应用实例全面了解开发过程《Cortex-M3 +  $\mu$  C/OS-II嵌入式系统开发入门与应用》介绍了Cortex—M3内核和在Cortex - M3平台上用pCOS操作系统开发的方法，内容涵盖：走进ARM微处理器Cortex—M3处理器体系结构ARM开发环境RealView MDK平台搭建基于RealView开发环境的嵌入式软件开发STM32F103处理器内部资源C编程与实例 $\mu$  COS-II操作系统基础及其移植开发初步 $\mu$  COS-II的内核机制基于 $\mu$  COS-II的程序设计实例

#### 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>