

21天学通

ARM嵌入式开发

(第2版)

• 薛园园 编著 •

15
小时多媒体
语音视频教学
DVD

本书特色

- 基础知识→核心技术→典型实例→综合练习→项目案例
- 330个典型实例、2个系统移植案例、75个问答题、117个面试题
- 一线开发人员全程贴心讲解，上手毫不费力

超值DVD

- 15小时多媒体语音视频教学
- 本书电子教案（PPT）



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

21天学编程系列

21天学编程系列

21天学通

ARM嵌入式开发

(第2版)

• 薛园园 编著 •

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书系统地介绍了 ARM 嵌入式系统的基本结构、开发和 ARM 处理器的编程,并结合流行的嵌入式系统介绍了如何移植。本书言简意赅、通俗易懂,知识点覆盖全面,详细讲述了最新的 ARM Cortex-M3 内核及 STM32 处理器编程,使读者能够更好地掌握 ARM 处理器及相应的嵌入式系统设计。

本书分为四篇,共 23 章,全面详细地讲述了 ARM 处理器的原理、程序设计方法、ARM 处理器编程指南及嵌入式系统移植。首先介绍了 ARM 处理器的构架及如何搭建一个 ARM 嵌入式开发平台,接着介绍了 ARM Cortex-M3 的指令集,基于 Cortex-M3 的 STM32 系列芯片及一些典型的嵌入式操作系统。然后结合 STM32 的各个功能部件详细讲解了 ARM 处理器的硬件编程方法。最后本书还介绍了基于 ARM Cortex-M3 的 STM32 芯片如何移植流行的嵌入式操作系统。

本书知识点覆盖全面、结构安排紧凑、讲解详细、实例丰富。ARM 嵌入式系统的初学者通过本书可以快速掌握程序设计方法。本书对具有一定开发经验的设计人员,也有很高的参考价值。

本书附赠 DVD 光盘 1 张,内容包括超大容量手把手教学视频、电子教案(PPT)。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

21 天学通 ARM 嵌入式开发 / 薛园园编著. —2 版. —北京: 电子工业出版社, 2011.9

(21 天学编程系列)

ISBN 978-7-121-14162-1

I. ①2… II. ①薛… III. ①微处理器, ARM—系统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字(2011)第 146583 号

责任编辑: 高洪霞

印 刷:

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 26.25 字数: 631 千字

印 次: 2011 年 9 月第 1 次印刷

印 数: 5000 册 定价: 49.80 元(含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

本书特点

千里之行，始于足下！

——老子

为什么要写这样一本书

嵌入式系统是目前最为流行的技术。在嵌入式系统中，ARM 处理器的应用无处不在。熟练掌握 ARM 处理器及嵌入式操作系统设计，已经成为设计人员的法宝。目前市面上关于 ARM 嵌入式开发的书籍非常丰富，但是这些书有些过于技术化，入门者根本无法入手，也有部分书籍内容不全，无法让初学者对该技术得到全面的认识。另外，这些书籍内容过于陈旧，无法跟踪最新的进展。

为了能让初次接触 ARM 嵌入式系统开发的爱好者快速而又轻松地学会 ARM 处理器及其程序开发，笔者总结了自己学习 ARM 的经验，并结合多年实际开发的经验，编写了这本 ARM 嵌入式开发的基础教程。在本书中，笔者从最基础的概念入手，循序渐进地将 ARM 处理器开发和程序设计中的每个技术点展现在读者面前，力求让读者在最短的时间内高效地掌握 ARM 处理器及嵌入式开发的基础概念及技术要点。本书采用了最新的 ARM Cortex-M3 内核进行讲解。最新的 ARM Cortex-M3 内核提供了更为强大的功能，目前正在取代老款的 ARM7 等内核。

本书有何特色

1. 细致体贴的讲解

为了让读者更快地上手，本书特别设计了适合初学者的学习方式，用准确的语言总结概念
■用直观的图示演示过程■用详细的注释解释代码■用形象的比方帮助记忆。效果如下：

① **知识点介绍** 准确、清晰是其显著特点，一般放在每一节开始位置，让零基础的读者了解相关概念，顺利入门。

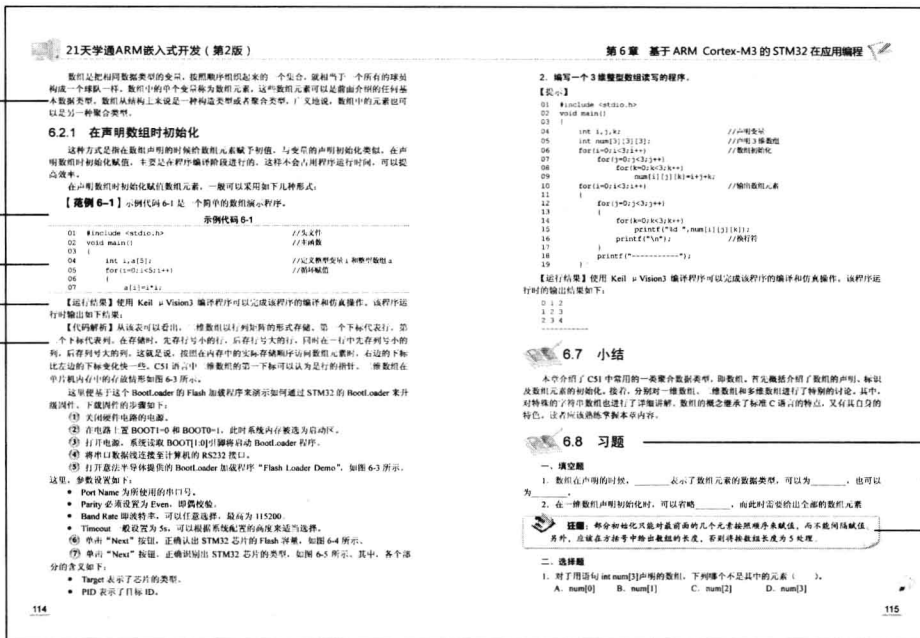
② **范例** 书中出现的完整实例，以章节顺序编号，便于检索和循序渐进地学习、实践，放在每节知识点介绍之后。

③ **范例代码** 与范例编号对应，层次清楚、语句简洁、注释丰富，体现了代码优美的原则，有利于读者养成良好的代码编写习惯。对于大段程序，均在每行代码前设定编号，便于学习。

④ **运行结果** 对范例给出运行结果和对应图示，帮助读者更直观地理解范例代码。

⑤ **代码解析** 将范例代码中的关键代码行逐一解释，有助于读者掌握相关概念和知识。

⑥ **习题** 每章最后提供专门的测试习题，供读者检验所学知识是否牢固掌握。



2. 编写一个 3 维整型数组读写程序。

【例 6-2】

```

01 #include <stdio.h>
02 void main()
03 {
04     int i,j,k;               //声明变量
05     int num[3][3][3];       //声明 3 维数组
06     for(i=0;i<3;i++)        //遍历初始化
07     {
08         for(j=0;j<3;j++)
09         {
10             for(k=0;k<3;k++)
11             {
12                 num[i][j][k]=i+j+k; //输出数组元素
13             }
14         }
15     }
16     for(k=0;k<3;k++)
17     {
18         for(i=0;i<3;i++)
19         {
20             for(j=0;j<3;j++)
21             {
22                 printf("%d\t",num[i][j][k]); //输出
23             }
24         }
25     }
26 }

```

【运行结果】使用 Keil uVision 编译程序可以完成该程序的编译和仿真操作。该程序运行时的输出结果如下：

```

0 1 2
1 2 3
2 3 4

```

6.7 小结

本章介绍了 C51 中常用的一些基本数据类型。即数组。首先就介绍了数组的声明、标识及数组元素的初始化。接着，分别对一维数组、二维数组和三维数组进行了详细的讨论。其中，对特殊的 7 种数组也进行了详细的讨论。数组的概念继承了标准 C 语言的特点，又有其自身的特色。这应该给读者带来新的启示。

6.8 习题

一、填空题

1. 数组在声明的时候，_____表示了数组元素的数据类型，可以为_____，也可以为_____。
2. 在一维数组声明初始化时，可以省略_____，而此时代码会出全部的数组元素。

二、选择题

1. 对于语句 int num[3] 声明的数组，下列哪个不是其中的元素 ()。

A. num[0] B. num[1] C. num[2] D. num[3]

⑦ 贴心的提示 为了便于读者阅读，全书还穿插着一些技巧、提示等小贴士，体例约定如下：

- 提示：通常是一些贴心的提醒，让读者加深印象或提供建议，或者解决问题的方法。
- 注意：提出学习过程中需要特别注意的一些知识点和内容，或者相关信息。
- 警告：对操作不当或理解偏差将会造成的灾难性后果做警示，以加深读者印象。

经作者多年的培训和授课证明，以上讲解方式是最适合初学者学习的方式，读者按照这种方式，会非常轻松、顺利地掌握本书知识。

2. 实用超值的 DVD 光盘

为了帮助读者比较直观地学习，本书附赠 DVD 光盘，内容包括多媒体视频、电子教案 (PPT) 等。

● 多媒体视频

配有长达 15 小时手把手教学视频，讲解关键知识点界面操作。作者亲自配音、演示，手把手教会读者使用。



● 电子教案（PPT）

本书可以作为高校相关课程的教材或课外辅导书，所以笔者特别为本书制作了电子教案（PPT），以方便老师教学使用。



3. 提供完善的技术支持

本书提供了论坛：<http://www.rzchina.net>，读者可以在上面提问交流。另外，论坛上还有一些小的教程、视频动画和各种技术文章，可帮助读者提高开发水平。

推荐的学习计划

本书作者在长期从事相关培训或教学实践过程中，归纳了最适合初学者的学习模式，并参考了多位专家的意见，为读者总结了合理的学习时间分配方式，列表如下：

推荐时间安排		自学目标（框内打钩表示已掌握）	难度指数	
第 1 周	第 1 天	了解嵌入式系统。	☐	★
		了解嵌入式系统的特点及发展趋势。	☐	
		掌握和熟练嵌入式系统的开发过程。	☐	
		如何成为嵌入式开发人员。	☐	
		了解嵌入式操作系统的分类和特点。	☐	
		掌握各种主流的嵌入式操作系统。	☐	
	第 2 天	了解 ARM 体系结构。	☐	★★
		掌握各种 ARM 处理器系列。	☐	
		掌握 ARM Cortex-M3 的处理器结构。	☐	
		掌握 ARM Cortex-M3 的编程模型。	☐	
		掌握典型的 ARM 开发调试工具。	☐	
		了解和掌握基于 ARM 的 RealView MDK 开发环境。	☐	
		了解基于 ARM Cortex-M3 的 STM32 芯片系列。	☐	
掌握基于 ARM Cortex-M3 的 STM32 最小开发系统。		☐		
第 3 天	掌握 STM32 的程序下载方式。	☐	★★★	
	了解和掌握 ARM Cortex-M3 的数据传送指令。	☐		
	了解和掌握 ARM Cortex-M3 的数据处理指令。	☐		
	了解和掌握子程序呼叫及转移指令。	☐		
	了解和掌握 ARM Cortex-M3 的隔离指令。	☐		
	了解 ARM Cortex-M3 所不支持的指令。	☐		

续表

推荐时间安排		自学目标（框内打钩表示已掌握）	难度指数
第 1 周	第 4 天	了解在应用编程。 ☐ 了解 STM32 的在应用编程及 BootLoader。 ☐ 掌握基本 STM32 的启动配置。 ☐ 掌握 STM32 的在应用编程。 ☐	★★★
	第 5 天	了解 FreeRTOS 的基本情况。 ☐ 了解嵌入式操作系统的基本概念。 ☐ 掌握 FreeRTOS 的 API 模块。 ☐	★★★★★
	第 6 天	了解 UC/OS- II 的基本情况。 ☐ 掌握 UC/OS- II 的任务管理。 ☐ 掌握 UC/OS- II 的任务间通信与同步方法。 ☐	★★★★★
	第 7 天	了解通用 I/O 接口的结构和功能。 ☐ 掌握复用 I/O 功能及其重映射。 ☐ 掌握和熟练通用 I/O 接口的程序设计方法。 ☐	★★★
第 2 周	第 8 天	了解通用 Flash 闪存的结构。 ☐ 了解通用 Flash 闪存的基本特性。 ☐ 掌握和熟练 Flash 闪存的程序设计方法。 ☐	★★★
	第 9 天	了解 ADC 系统的引脚结构。 ☐ 掌握 ADC 系统的功能特点。 ☐ 掌握和熟练通用 ADC 系统的程序设计方法。 ☐	★★★★★
	第 10 天	掌握嵌套向量中断控制器（NVIC）。 ☐ 掌握和熟练嵌套向量中断控制器（NVIC）的编程方法。 ☐ 掌握外部中断/事件控制器（EXTI）。 ☐ 掌握和熟练外部中断/事件控制器（EXTI）的编程方法。 ☐	★★★
	第 11 天	掌握 USART 的结构。 ☐ 掌握和熟练 USART 的编程方法。 ☐ 掌握 CAN 的结构。 ☐ 掌握和熟练 CAN 的编程方法。 ☐	★★★★★
	第 12 天	掌握 SPI 的结构。 ☐ 掌握和熟练 SPI 的编程方法。 ☐ 掌握 I2C 的结构。 ☐ 掌握和熟练 I2C 的编程方法。 ☐	★★★★★
	第 13 天	掌握和熟练系统计时器（SysTick）的程序设计方法。 ☐ 了解通用定时器（TIM）。 ☐ 掌握和熟练通用定时器（TIM）的程序设计方法。 ☐	★★★★★
	第 14 天	了解高级控制定时器（TIM1）。 ☐ 掌握和熟练高级控制定时器（TIM1）的程序设计方法。 ☐	★★★★★
第 3 周	第 15 天	掌握复位和时钟控制（RCC）。 ☐ 掌握和熟练时钟控制（RCC）的程序设计方法。 ☐ 掌握实时时钟（RTC）。 ☐ 掌握和熟练实时时钟（RTC）的程序设计方法。 ☐	★★★★★

续表

推荐时间安排		自学目标（框内打钩表示已掌握）	难度指数
第3周	第16天	了解独立看门狗（IWDG）的功能。 ☐	★★★★
		掌握和熟练独立看门狗（IWDG）的程序设计方法。 ☐	
		了解窗口看门狗（WWDG）的功能。 ☐	
		掌握和熟练窗口看门狗（WWDG）的程序设计方法。 ☐	
	第17天	了解备份寄存器（BKP）的功能。 ☐	★★
		掌握和熟练备份寄存器（BKP）的程序设计方法。 ☐	
	第18天	了解 DMA 控制器的基本特性。 ☐	★★★★
		掌握 DMA 控制器的基本功能。 ☐	
		掌握和熟练 DMA 控制器的程序设计方法。 ☐	
	第19天	了解电源控制的结构。 ☐	★★★
		掌握 STM32 的电源管理器功能。 ☐	
		掌握 STM32 的低功耗模式。 ☐	
		掌握和熟练电源控制（PWR）的程序设计方法。 ☐	
	第20天	了解如何获取 FreeRTOS 内核。 ☐	★★★★
		掌握 FreeRTOS 在 STM32 上的移植。 ☐	
		掌握 FreeRTOS 的任务创建。 ☐	
	第21天	了解如何获取 UC/OS-II 嵌入式系统内核。 ☐	★★★★
		掌握 UC/OS-II 的任务管理。 ☐	
		掌握 UC/OS-II 的任务间通信与同步方法。 ☐	

本书适合哪些读者阅读

本书非常适合以下人员阅读：

- 从未接触过 ARM 开发技术的人员；
- 有一定硬件开发技术基础，但还需要进一步学习的人员；
- 想学习 ARM 嵌入式编程的开发人员；
- 其他编程爱好者。

目 录

第一篇 走进嵌入式系统开发

第1章 嵌入式系统开发概述

( 教学视频: 44 分钟) 25

- 1.1 何为嵌入式系统 25
 - 1.1.1 嵌入式系统的概念 25
 - 1.1.2 嵌入式系统的基本结构 26
 - 1.1.3 嵌入式处理器 26
- 1.2 嵌入式系统的特点及发展趋势 27
- 1.3 嵌入式系统的开发过程 28
 - 1.3.1 芯片选择 28
 - 1.3.2 配置硬件平台 28
 - 1.3.3 BootLoader 移植 28
 - 1.3.4 操作系统裁减 29
 - 1.3.5 系统移植 29
 - 1.3.6 应用程序开发 29
- 1.4 如何成为嵌入式开发人员 29
- 1.5 小结 30
- 1.6 习题 30

第2章 嵌入式操作系统简介

( 教学视频: 37 分钟) 31

- 2.1 嵌入式操作系统概述 31
 - 2.1.1 嵌入式操作系统的发展历史 31
 - 2.1.2 嵌入式操作系统的分类 31
 - 2.1.3 嵌入式操作系统的特点 32
- 2.2 主流嵌入式操作系统简介 32
 - 2.2.1 VxWorks 32
 - 2.2.2 pSOS 33
 - 2.2.3 Palm OS 33
 - 2.2.4 QNX 34
 - 2.2.5 OS-9 34
 - 2.2.6 LynxOS 34
 - 2.2.7 Windows CE 34
 - 2.2.8 μ C/OS-II 35
 - 2.2.9 嵌入式 Linux 36
 - 2.2.10 FreeRTOS 36
- 2.3 小结 36
- 2.4 习题 36

第3章 ARM 处理器构架

( 教学视频: 53 分钟) 38

- 3.1 ARM 体系结构概述 38

- 3.1.1 ARM 简介 38
- 3.1.2 CISC 的体系结构 39
- 3.1.3 RISC 的体系结构 39
- 3.1.4 ARM 微处理器的应用领域 39
- 3.1.5 ARM 体系结构的特点 40
- 3.2 ARM 处理器系列 40
 - 3.2.1 ARM7 微处理器 40
 - 3.2.2 ARM9 微处理器 41
 - 3.2.3 ARM9E 微处理器 41
 - 3.2.4 ARM10E 微处理器 42
 - 3.2.5 ARM11 微处理器 42
 - 3.2.6 SecurCore 微处理器 42
 - 3.2.7 StrongARM 微处理器 42
 - 3.2.8 Xscale 微处理器 43
 - 3.2.9 Cortex-M3 微处理器 43
 - 3.2.10 基于 ARM 核的芯片选择 43
- 3.3 ARM 处理器结构 44
 - 3.3.1 ARM 的系统构架 44
 - 3.3.2 ARM 的流水线结构 44
 - 3.3.3 ARM 的存储器结构 45
- 3.4 ARM 编程模型 45
 - 3.4.1 工作模式和工作状态 46
 - 3.4.2 特权访问和用户访问 46
 - 3.4.3 数据类型 46
 - 3.4.4 存储器格式 46
- 3.5 ARM 开发调试工具综述 47
- 3.6 小结 47
- 3.7 习题 47

第4章 搭建嵌入式开发平台

( 教学视频: 39 分钟) 49

4.1 基于 ARM 的 RealView MDK 开发

- 环境 49
 - 4.1.1 RealView MDK 简介 49
 - 4.1.2 RealView MDK 系统安装需求 49
 - 4.1.3 RealView MDK 的安装 50
 - 4.1.4 RealView MDK 集成开发环境总览 51
 - 4.1.5 RealView MDK 的菜单栏 51
 - 4.1.6 RealView MDK 的工具栏 55
 - 4.1.7 RealView MDK 的快捷键 58

4.2 基于 ARM Cortex-M3 的 STM32 选型表	58	4.4 程序下载方式	60
4.3 基于 ARM Cortex-M3 的 STM32 最小开发系统	59	4.5 小结	61
		4.6 习题	61

第二篇 ARM 嵌入式系统开发基础

第 5 章 ARM CORTEX-M3 指令集

( 教学视频: 30 分钟) 62

5.1 数据传送指令	62
5.1.1 寄存器间数据传送指令	62
5.1.2 存储器数据传送指令	62
5.1.3 索引数据传送指令	63
5.2 数据处理指令	64
5.2.1 算术四则运算指令	64
5.2.2 逻辑及移位操作指令	65
5.2.3 数据序转及位操作指令	66
5.3 子程序呼叫及转移指令	66
5.4 隔离指令	67
5.5 不支持的指令	67
5.6 小结	68
5.7 习题	68

第 6 章 基于 ARM CORTEX-M3 的 STM32 在应用编程

( 教学视频: 31 分钟) 70

6.1 在应用编程简介	70
6.2 STM32 的启动配置	70
6.3 STM32 的在应用编程	71
6.3.1 电路原理图	71
6.3.2 BootLoader 固件下载步骤	72
6.4 小结	74
6.5 习题	75

第 7 章 嵌入式实时操作系统 FREERTOS

( 教学视频: 42 分钟) 76

7.1 嵌入式实时操作系统 FreeRTOS 简介	76
7.2 嵌入式操作系统基本概念	76
7.2.1 多任务	77

7.2.2 调度	77
7.2.3 上下文切换	78
7.2.4 实时应用和调度	78
7.3 FreeRTOS 的 API 模块	78
7.3.1 定制	78
7.3.2 任务创建和删除	80
7.3.3 任务控制	81
7.3.4 内核控制	82
7.4 小结	83
7.5 习题	83

第 8 章 嵌入式系统 UC/OS-II

( 教学视频: 39 分钟) 85

8.1 UC/OS-II 嵌入式系统简介	85
8.2 UC/OS-II 任务管理	85
8.2.1 建立任务 OSTaskCreate	86
8.2.2 建立任务 OSTaskCreate Ext	86
8.2.3 任务堆栈	88
8.2.4 堆栈检验 OSTaskStkChk	88
8.2.5 删除任务 OSTaskDel	89
8.2.6 请求删除任务 OSTaskDelReq	90
8.2.7 改变任务的优先级 OSTaskChangePrio	91
8.2.8 挂起任务 OSTaskSuspend	92
8.2.9 恢复任务 OSTaskResume	93
8.2.10 获得有关任务的信息 OSTaskQuery	94
8.3 任务之间的通信与同步	95
8.3.1 信号量	95
8.3.2 邮箱	95
8.3.3 消息队列	95
8.4 小结	96
8.5 习题	96

第三篇 深入 ARM 嵌入式控制器

第 9 章 通用 I/O 接口

( 教学视频: 48 分钟) 98

9.1 通用 I/O 接口的结构	98
9.1.1 GPIO 的基本结构	98
9.1.2 GPIO 的功能	99

9.1.3 GPIO 的功能配置	100
9.1.4 复用功能与重映射	101
9.2 通用 I/O 接口的编程方法	104
9.2.1 GPIO 寄存器结构	105
9.2.2 GPIO 寄存器复位函数 GPIO_DeInit	105

9.2.3	复用功能寄存器复位函数 GPIO_AFIODeInit	106	10.3.6	锁定擦除控制器函数 Flash_Lock.....	119
9.2.4	参数初始化函数 GPIO_Init	106	10.3.7	擦除页函数 Flash_ErasePage	119
9.2.5	复位 GPIO_InitStruct 函数 GPIO_StructInit.....	108	10.3.8	擦除全部页函数 Flash_EraseAllPages	119
9.2.6	读输入引脚函数 GPIO_ReadInputDataBit	108	10.3.9	擦除选项字节函数 Flash_EraseOptionBytes	120
9.2.7	读输入端口函数 GPIO_ReadInputData	108	10.3.10	字写入函数 Flash_ProgramWord.....	120
9.2.8	读输出引脚函数 GPIO_ReadOutputDataBit	109	10.3.11	半字写入函数 Flash_ProgramHalfWord	121
9.2.9	读输出端口函数 GPIO_ReadOutputData	109	10.3.12	写选项字节函数 Flash_ ProgramOptionByteData	121
9.2.10	置位引脚函数 GPIO_SetBits.....	109	10.3.13	页写保护函数 Flash_ EnableWriteProtection.....	122
9.2.11	清除引脚位函数 GPIO_ResetBits.....	109	10.3.14	使能或禁用读保护函数 Flash_ReadOutProtection	123
9.2.12	置位或清除引脚函数 GPIO_WriteBit.....	110	10.3.15	写选项字节函数 Flash_ UserOptionByteConfig.....	124
9.2.13	写端口函数 GPIO_Write.....	110	10.3.16	获取选项字节函数 Flash_ GetUserOptionByte	125
9.2.14	引脚配置锁定函数 GPIO_PinLockConfig.....	110	10.3.17	获取写保护选项字节函数 Flash_ GetWriteProtectionOptionByte	125
9.2.15	选择事件输出引脚函数 GPIO_EventOutputConfig	111	10.3.18	获取读保护状态函数 Flash_Get ReadOutProtectionStatus	125
9.2.16	使能事件输出函数 GPIO_EventOutputCmd	111	10.3.19	获取预取缓冲区状态函数 Flash_GetPrefetchBufferStatus	125
9.2.17	改变引脚映射函数 GPIO_PinRemapConfig.....	111	10.3.20	使能或禁用中断函数 Flash_ITConfig.....	126
9.2.18	选择 EXTI 线函数 GPIO_EXTILineConfig.....	112	10.3.21	获取标志位状态函数 Flash_GetFlagStatus.....	126
9.3	综合练习.....	113	10.3.22	清除挂起标志函数 Flash_ClearFlag	127
9.4	小结	113	10.3.23	获取 Flash 状态函数 Flash_GetStatus	127
9.5	习题	114	10.3.24	等待操作结束函数 Flash_ WaitForLastOperation	127
第 10 章 FLASH			10.4	综合练习	128
( 教学视频: 24 分钟)			10.5	小结	130
10.1	Flash 闪存的结构	115	10.6	习题	130
10.2	Flash 闪存的基本特性	116	第 11 章 ADC 系统		
10.2.1	Flash 闪存读取	116	( 教学视频: 52 分钟)		
10.2.2	Flash 闪存编程和擦除	116	11.1	ADC 系统概述	132
10.3	Flash 的编程方法	116	11.2	ADC 系统功能特性	133
10.3.1	Flash 寄存器结构	117	11.2.1	ADC 开关控制	133
10.3.2	设置代码延时值函数 Flash_SetLatency.....	118	11.2.2	ADC 时钟	133
10.3.3	使能或禁用半周期访问函数 Flash_HalfCycleAccessCmd	118	11.2.3	ADC 通道选择	133
10.3.4	使能或禁用预取缓冲区函数 Flash_PrefetchBufferCmd	118			
10.3.5	解锁擦除控制器函数 Flash_Unlock.....	118			

11.2.4	ADC 的转换模式	133	11.3.17	使能或禁用外部触发函数	
11.2.5	中断	134		ADC_ExternalTrigConvCmd	148
11.2.6	模拟看门狗	134	11.3.18	获取转换结果函数 ADC_	
11.2.7	ADC 的扫描模式	134		GetConversionValue	148
11.2.8	ADC 的注入通道管理	134	11.3.19	获取双重转换结果函数 ADC_	
11.2.9	间断模式	135		GetDualModeConversionValue	148
11.2.10	ADC 的校准模式	135	11.3.20	使能或禁用自动注入组转换函数	
11.2.11	ADC 的数据对齐	136		ADC_AutoInjectedConvCmd	149
11.2.12	可编程的通道采样时间	136	11.3.21	使能或禁用非连续模式函数	
11.2.13	外部触发转换	136		ADC_InjectedDiscModeCmd	149
11.2.14	DMA 请求	137	11.3.22	配置外部触发器函数 ADC_	
11.2.15	双 ADC 模式	137		ExternalTrigInjectedConvConfig	149
11.2.16	温度传感器	138	11.3.23	使能或禁用注入信道转换	
11.3	ADC 系统的编程方法	139		函数 ADC_ExternalTrig	
11.3.1	ADC 寄存器结构	140		InjectedConvCmd	150
11.3.2	ADC 寄存器复位函数 ADC_		11.3.24	使能或禁用开始注入信道	
	DeInit	141		转换函数 ADC_Software	
11.3.3	参数初始化函数 ADC_Init	141		StartInjectedConvCmd	150
11.3.4	复位 ADC_InitStruct 函数		11.3.25	获取启动转换注入状态	
	ADC_StructInit	143		函数 ADC_GetSoftware	
11.3.5	使能或禁用 ADC 模块函数			StartInjectedConvStatus	151
	ADC_Cmd	143	11.3.26	配置 ADC 注入信道函数	
11.3.6	使能或禁用 DMA 请求函数			ADC_InjectedChannelConfig	151
	ADC_DMAMCmd	143	11.3.27	配置注入信道音序器长度	
11.3.7	使能或禁用 ADC 中断函数			函数 ADC_Injected	
	ADC_ITConfig	144		SequencerLengthConfig	152
11.3.8	复位 ADC 校准寄存器函数		11.3.28	设置注入信道转换值位移函数	
	ADC_ResetCalibration	144		ADC_SetInjectedOffset	152
11.3.9	获取复位校准寄存器状态函数 ADC_		11.3.29	获取注入信道转换结果	
	GetResetCalibrationStatus	144		函数 ADC_GetInjected	
11.3.10	开始 ADC 校准函数			ConversionValue	153
	ADC_StartCalibration	145	11.3.30	使能或禁用模拟看门狗函数	
11.3.11	获取 ADC 校准状态			ADC_AnalogWatchdogCmd	153
	ADC_GetCalibrationStatus	145	11.3.31	配置模拟看门狗域值函数	
11.3.12	使能或禁用软件转换函数 ADC_			ADC_AnalogWatchdog	
	SoftwareStartConvCmd	145		ThresholdsConfig	154
11.3.13	获取软件开始转换状态函数 ADC_		11.3.32	配置模拟看门狗保护函数	
	GetSoftwareStartConvStatus	146		ADC_AnalogWatchdog	
11.3.14	配置常规组非连续模式函数 ADC_			SingleChannelConfig	154
	DiscModeChannelCountConfig	146	11.3.33	使能或禁用温度传感函数	
11.3.15	使能或禁用常规组非连续模式			ADC_TempSensorVrefintCmd	154
	函数 ADC_DiscModeCmd	146	11.3.34	获取 ADC 标志函数	
11.3.16	配置 ADC 常规信道函数 ADC_			ADC_GetFlagStatus	155
	RegularChannelConfig	147	11.3.35	清除挂起标志函数	
				ADC_ClearFlag	155

11.3.36	获取中断函数 ADC_GetITStatus	155
11.3.37	清除中断挂起位函数 ADC_ClearITPendingBit	156
11.4	综合练习	156
11.5	小结	157
11.6	习题	157
第 12 章 中断系统		
	(教学视频: 32 分钟)	159
12.1	嵌套向量中断控制器 (NVIC)	159
12.2	NVIC 的编程方法	161
12.2.1	NVIC 寄存器结构	162
12.2.2	复位 NVIC 外设函数 NVIC_DeInit	163
12.2.3	复位 SCB 外设函数 NVIC_ SCBDeInit	163
12.2.4	配置优先级组函数 NVIC_ PriorityGroupConfig	163
12.2.5	初始化 NVIC 外设函数 NVIC_Init	163
12.2.6	复位 NVIC_InitStruct 函数 NVIC_StructInit	164
12.2.7	使能 PRIMASK 优先级函数 NVIC_SETPRIMASK	165
12.2.8	用 PRIASK 优先级函数 NVIC_RESETPRIMASK	165
12.2.9	使能 FAULTMASK 优先级函数 NVIC_SETFAULTMASK	165
12.2.10	禁用 FAULTMASK 优先级函数 NVIC_RESETFAULTMASK	165
12.2.11	配置执行优先级函数 NVIC_ BASEPRICONFIG	165
12.2.12	获取 BASEPRI 掩码值函数 NVIC_GetBASEPRI	166
12.2.13	获取挂起服务的 IRQ 通道 标识符函数 NVIC_GetCurrent PendingIRQChannel	166
12.2.14	获取 IRQ 通道挂起位函数 NVIC_ GetIRQChannelPendingBitStatus	166
12.2.15	设置中断挂起位函数 NVIC_ GetIRQChannelPendingBit	166
12.2.16	清除中断挂起位函数 NVIC_ ClearIRQChannelPendingBit	167
12.2.17	返回活动句柄函数 NVIC_ GetCurrentActiveHandle	167
12.2.18	获取 IRQ 通道活动位函数 NVIC_ GetIRQChannelActiveBitStatus	167
12.2.19	获取 ID 号函数 NVIC_GetCPUID	167
12.2.20	设置向量表和偏移量函数 NVIC_SetVectorTable	168
12.2.21	系统复位函数 NVIC_ GenerateSystemReset	168
12.2.22	核复位函数 NVIC_GenerateCoreReset	168
12.2.23	低功耗模式条件函数 NVIC_SystemLPConfig	168
12.2.24	使能或禁用系统句柄函数 NVIC_SystemHandlerConfig	169
12.2.25	配置系统句柄优先级函数 NVIC_ SystemHandlerPriorityConfig	169
12.2.26	获取系统句柄挂起位函数 NVIC_GetSystemHandler PendingBitStatus	170
12.2.27	设置系统句柄挂起位函数 NVIC_ SetSystemHandlerPendingBit	170
12.2.28	清除系统句柄挂起位函数 NVIC_ ClearSystemHandlerPendingBit	171
12.2.29	获取系统句柄活动位函数 NVIC_GetSystemHandler ActiveBitStatus	171
12.2.30	获取系统故障句柄源函数 NVIC_ GetFaultHandlerSources	172
12.2.31	获取故障句柄位置函数 NVIC_GetFaultAddress	172
12.3	外部中断/事件控制器 (EXTI)	173
12.3.1	EXTI 主要特性	173
12.3.2	EXTI 功能说明	173
12.4	EXTI 的编程方法	174
12.4.1	EXTI 寄存器结构	174
12.4.2	复位 EXTI 外设函数 EXTI_DeInit	175
12.4.3	初始化 EXTI 外设函数 EXTI_Init	175
12.4.4	复位 EXTI_InitStruct 函数 EXTI_StructInit	176
12.4.5	软中断函数 EXTI_GenerateSWInterrupt	176
12.4.6	获取标志状态函数 EXTI_GetFlagStatus	176

12.4.7	清除挂起标志函数 EXTI_ClearFlag	176	13.2.12	使能或者禁用 LIN 模式 函数 USART_LINCmd	195
12.4.8	获取 EXTI 置位函数 EXTI_GetITStatus	177	13.2.13	发送数据函数 USART_ SendData	195
12.4.9	清除挂起位函数 EXTI_ClearITPendingBit	177	13.2.14	接收数据函数 USART_ ReceiveData	196
12.5	综合练习	177	13.2.15	传输间断字符函数 USART_SendBreak	196
12.6	小结	181	13.2.16	设置预警时间函数 USART_SetGuardTime	196
12.7	习题	181	13.2.17	设置时钟预分频器函数 USART_SetPrescaler	196
第 13 章 异步串行通信接口			13.2.18	使能或者禁用智能卡模式函数 USART_SmartCardCmd	197
(教学视频: 59 分钟)			13.2.19	使能或禁用 NACK 函数 USART_SmartCardNackCmd	197
13.1	USART 结构	184	13.2.20	使能或者禁用半双工模式函数 USART_HalfDuplexCmd	197
13.1.1	USART 特征描述	184	13.2.21	配置 IrDA 模式函数 USART_ IrDAConfig	198
13.1.2	USART 发送器	184	13.2.22	使能或者禁用 IrDA 模式 函数 USART_IrDACmd	198
13.1.3	USART 接收器	185	13.2.23	获取标志是否置位函数 USART_GetFlagStatus	198
13.1.4	LIN 模式	185	13.2.24	清除未决标志函数 USART_ClearFlag	199
13.1.5	USART 同步模式	186	13.2.25	获取中断状态函数 USART_GetITStatus	199
13.1.6	USART 单线半双工通信	186	13.2.26	清除中断未决位函数 USART_ ClearITPendingBit	200
13.1.7	USART 的智能卡功能	186	13.3	CAN 结构	200
13.1.8	USART 的 IrDA 模式	186	13.3.1	初始化模式	201
13.1.9	USART 的 DMA 通信	187	13.3.2	正常模式	201
13.1.10	USART 的硬件流控制	188	13.3.3	睡眠模式	201
13.1.11	USART 的中断请求	188	13.3.4	测试模式	202
13.2	USART 编程方法	188	13.3.5	静默模式	202
13.2.1	USART 寄存器结构	189	13.3.6	环回模式	202
13.2.2	USART 寄存器复位函数 USART_DeInit	190	13.3.7	环回静默模式	202
13.2.3	参数初始化函数 USART_Init	190	13.4	CAN 编程方法	202
13.2.4	复位 USART_InitStruct 函数 USART_StructInit	192	13.4.1	CAN 寄存器结构	203
13.2.5	使能或禁用 USART 函数 USART_Cmd	192	13.4.2	CAN 寄存器复位函数 CAN_DeInit	204
13.2.6	使能或禁用中断函数 USART_ITConfig	193	13.4.3	参数初始化函数 CAN_Init	204
13.2.7	使能或禁用 DMA 函数 USART_DMACmd	193	13.4.4	过滤参数初始化函数 CAN_FilterInit	207
13.2.8	设置节点地址函数 USART_SetAddress	194			
13.2.9	选择唤醒方式函数 USART_WakeUpConfig	194			
13.2.10	USART 模式函数 USART_ ReceiverWakeUpCmd	194			
13.2.11	LIN 间隔检波长度函数 USART_ LINBreakDetectionConfig	195			

13.4.5	复位 CAN_InitStruct 函数 CAN_StructInit	208
13.4.6	使能或禁用 CAN 中断函数 CAN_ITConfig	209
13.4.7	初始化报文发送函数 CAN_Transmit	209
13.4.8	检查报文发送函数 CAN_TransmitStatus	211
13.4.9	取消发送函数 CAN_ CancelTransmit	211
13.4.10	释放 FIFO 函数 CAN_ FIFORelease	212
13.4.11	获取挂起报文数量函数 CAN_MessagePending	212
13.4.12	报文接收函数 CAN_Receive	212
13.4.13	低功耗模式函数 CAN_Sleep	213
13.4.14	唤醒 CAN 模块函数 CAN_WakeUp	214
13.4.15	获取 CAN 标志函数 CAN_GetFlagStatus	214
13.4.16	清除挂起标志 CAN_ClearFlag	214
13.4.17	获取中断状态函数 CAN_GetITStatus	215
13.4.18	清除中断挂起位函数 CAN_ClearITPendingBit	216
13.5	综合练习	217
13.6	小结	218
13.7	习题	219

第 14 章 同步串行通信接口

	( 教学视频: 52 分钟)	221
14.1	SPI 结构	221
14.1.1	SPI 从模式工作原理	222
14.1.2	SPI 主模式工作原理	222
14.1.3	SPI 通信的 CRC	223
14.1.4	SPI 的 DMA 通信	223
14.1.5	SPI 的中断请求	223
14.2	SPI 编程方法	224
14.2.1	SPI 寄存器结构	224
14.2.2	复位 SPI 外设寄存器函数 SPI_DeInit	225
14.2.3	初始化外设 SPIx 寄存器 函数 SPI_Init	225
14.2.4	设置 SPI_InitStruct 结构 成员函数 SPI_StructInit	226

14.2.5	使能或者禁用 SPI 外设 函数 SPI_Cmd	226
14.2.6	使能或者禁用 SPI 中断函数 SPI_ITConfig	226
14.2.7	使能或者禁用 SPI 的 DMA 请求函数 SPI_DMAMCmd	227
14.2.8	发送数据函数 SPI_SendData	227
14.2.9	接收数据函数 SPI_Receive Data	227
14.2.10	配置 NSS 引脚函数 SPI_ NSSInternalSoftwareConfig	227
14.2.11	使能或者禁用 SPI SS 输出 函数 SPI_SSOutputCmd	228
14.2.12	设置 SPI 数据大小函数 SPI_DataSizeConfig	228
14.2.13	发送 CRC 值函数 SPI_TransmitCRC	228
14.2.14	使能或者禁用 CRC 值计算 函数 SPI_CalculateCRC	228
14.2.15	获取 CRC 寄存器值函数 SPI_GetCRC	229
14.2.16	获取 CRC 多项式寄存器值函数 SPI_GetCRCPolynomial	229
14.2.17	双向模式下数据方向函数 SPI_ BiDirectionalLineConfig	229
14.2.18	获取 SPI 标志位函数 SPI_GetFlagStatus	230
14.2.19	清除挂起标志位函数 SPI_ClearFlag	230
14.2.20	获取 SPI 中断函数 SPI_GetITStatus	230
14.2.21	清除中断挂起位函数 SPI_ClearITPendingBit	231
14.3	I2C 结构	231
14.3.1	I2C 从模式工作原理	232
14.3.2	I2C 主模式工作原理	233
14.3.3	SMBus 功能	233
14.3.4	I2C 的 DMA 功能	233
14.3.5	I2C 的中断请求	234
14.4	I2C 编程方法	235
14.4.1	I2C 寄存器结构	236
14.4.2	复位 I2C 外设寄存器函数 I2C_DeInit	236

14.4.3	初始化 I2C 外设函数 I2C_Init	237
14.4.4	复位 I2C_InitStruct 结构 函数 I2C_StructInit	238
14.4.5	使能或者禁用 I2C 外设 函数 I2C_Cmd	238
14.4.6	使能或者禁用 DMA 请求 函数 I2C_DMAMCmd	238
14.4.7	标示最后 DMA 传输函数 I2C_DMALastTransferCmd	239
14.4.8	产生起始信号函数 I2C_GenerateSTART	239
14.4.9	产生停止信号函数 I2C_GenerateSTOP	239
14.4.10	使能或者禁用应答函数 I2C_AcknowledgeConfig	240
14.4.11	配置 I2C 首地址 2 函数 I2C_OwnAddress2Config	240
14.4.12	使能或者禁用双重地址模式 函数 I2C_DualAddressCmd	240
14.4.13	使能或者禁用通用调用特性 函数 I2C_GeneralCallCmd	241
14.4.14	使能或者禁用 I2C 中断 函数 I2C_ITConfig	241
14.4.15	发送数据函数 I2C_SendData	241
14.4.16	接收数据函数 I2C_ReceiveData	242
14.4.17	发送地址字节函数 I2C_Send7bitAddress	242
14.4.18	读取 I2C 寄存器函数 I2C_ReadRegister	242
14.4.19	使能或禁用软件复位函数 I2C_SoftwareResetCmd	242
14.4.20	配置驱动 SMBAlert 引脚函数 I2C_SMBusAlertConfig	243
14.4.21	使能或者禁用 PEC 传输 函数 I2C_TransmitPEC	243
14.4.22	选择 PEC 位置函数 I2C_PECPositionConfig	243
14.4.23	使能或禁用 PEC 计算值 函数 I2C_CalculatePEC	244
14.4.24	获取 PEC 值函数 I2C_GetPEC	244
14.4.25	使能或者禁用 I2C ARP 函数 I2C_ARPCmd	244

14.4.26	使能或禁用 I2C 时钟伸展函数 I2C_StretchClockCmd	245
14.4.27	配置占空比函数 I2C_FastMode DutyCycleConfig	245
14.4.28	获取最近的 I2C 事件 函数 I2C_GetLastEvent	245
14.4.29	检查 I2C 事件函数 I2C_CheckEvent	246
14.4.30	获取 I2C 标志位函数 I2C_GetFlagStatus	246
14.4.31	清除挂起标志函数 I2C_ClearFlag	246
14.4.32	获取 I2C 中断函数 I2C_GetITStatus	247
14.4.33	清除中断挂起位函数 I2C_ClearITPendingBit	247
14.5	综合练习	248
14.6	小结	250
14.7	习题	251

第 15 章 通用定时器系统

( 教学视频: 39 分钟) ...253

15.1	系统计时器 (SysTick) 及其 编程方法	253
15.1.1	SysTick 寄存器结构	253
15.1.2	配置 SysTick 时钟源函数 SysTick_CLKSourceConfig	253
15.1.3	设置 SysTick 重载值函数 SysTick_Setreload	254
15.1.4	使能或禁用 SysTick 计数器 函数 SysTick_CounterCmd	254
15.1.5	使能或者取消 SysTick 中断 函数 SysTick_ITConfig	254
15.1.6	获取 SysTick 计数器值函数 SysTick_GetCounter	255
15.1.7	获取 SysTick 标志位函数 SysTick_GetFlagStatus	255
15.2	通用定时器 (TIM) 结构	255
15.2.1	时基单元	256
15.2.2	计数器的模式	256
15.2.3	通用定时器的特殊工作 模式	257
15.3	通用定时器 (TIM) 的编程方法	258
15.3.1	TIM 寄存器结构	260
15.3.2	复位 TIM 外设寄存器函数 TIM_DeInit	261