

高等院校信息技术规划教材

嵌入式系统原理及应用

——基于ARM Cortex-M3内核 的STM32F103系列微控制器

王益涵 孙宪坤 史志才 编著



清华大学出版社

高等院校信息技术规划教材

嵌入式系统原理及应用

——基于ARM Cortex-M3内核 的STM32F103系列微控制器

王益涵 孙宪坤 史志才 编著

清华大学出版社
北京

内 容 简 介

本书通过与常见的桌面通用系统比较,引入嵌入式系统的基本概念,主要介绍目前最新的 ARM Cortex-M3 内核以及意法半导体公司推出的基于 ARM Cortex-M3 内核的 STM32F103 微控制器。

全书共分 3 篇:第 1 篇(第 1、2 章)为系统篇,介绍嵌入式系统及其开发的基本概念;第 2 篇(第 3、4 章)为内核篇,分析 ARM Cortex-M3 内核以及基于 ARM Cortex-M3 内核的 STM32F103 微控制器的体系结构、工作原理、编程模型和开发方法等;第 3 篇(第 5~12 章)为片内外设篇,基于 STM32F103 微控制器讲述常用的片上外设/接口,包括 GPIO、定时器、EXTI、DMA、ADC、USART、SPI 和 I2C 等,并分别给出在 KEIL MDK 下采用库函数方式使用这些片上外设/接口进行应用开发的典型案例。

本书适合作为高等院校计算机、自动化、电子信息等电气信息类专业高年级本科生、研究生嵌入式相关课程的教材,同时可供从事嵌入式开发的技术和研究人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式系统原理及应用:基于 ARM Cortex-M3 内核的 STM32F103 系列微控制器/王益涵,孙宪坤,史志才编著. —北京:清华大学出版社,2016

高等院校信息技术规划教材

ISBN 978-7-302-44135-9

I. ①嵌… II. ①王… ②孙… ③史… III. ①单片微型计算机—高等学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2016)第 139130 号

责任编辑:焦虹 战晓雷

封面设计:傅瑞学

责任校对:梁毅

责任印制:何芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印刷者:北京富博印刷有限公司

装订者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:38.5

字 数:913千字

版 次:2016年10月第1版

印 次:2016年10月第1次印刷

印 数:1~2000

定 价:89.00元

前言

foreword

随着计算机和微电子技术的发展,作为面向特定应用而定制的专用计算机系统,嵌入式系统从8位、16位处理器时代跨入32位处理器时代,性能得到了显著提升,片上资源更加丰富,功能也越来越复杂和完善。尤其是进入21世纪以来,嵌入式系统因其具有体积小、功耗低、成本低、可靠实时等特点,深入人们生活的各个角落。大到国防军事、工业控制,小到消费电子、办公自动化,嵌入式系统无处不在。

随着嵌入式系统在各个行业的广泛应用,社会对嵌入式技术人才的需求量也日趋上升,具有一定开发经验的嵌入式工程师成为职场上的紧缺人才。目前,国内大多数高校的电气信息类专业都开设了嵌入式相关课程,社会上也有各种嵌入式培训班,以满足嵌入式人才培养的需求。但是,现有嵌入式系统书籍,或重“共性”——阐述嵌入式理论知识和基本原理,或重“特性”——讲解某款嵌入式处理器、某种嵌入式操作系统的原理及其应用开发;而缺乏一本“共性”和“个性”兼顾——既能较为系统地介绍嵌入式系统的基本概念和一般原理,又能指导初学者在实际软硬件环境中进行开发实践的嵌入式书籍。

针对上述情况,作者根据多年的嵌入式系统教学和开发经验,将嵌入式系统的理论知识和基于ARM Cortex-M3内核的STM32F103微控制器的实际开发相结合,编写了本书。

本书从结构上分为三大部分:

第一部分(第1、2章)为系统篇,通过与桌面系统的比较,介绍嵌入式系统及其开发的基本概念。

第二部分(第3、4章)为内核篇,分析目前最新的ARM Cortex-M3内核以及基于ARM Cortex-M3内核的STM32F103微控制器的体系结构、工作原理、编程模型和开发方法等。

第三部分(第5~12章)为片内外设篇,基于STM32F103微控制器讲述各个常用的片上外设/接口,包括GPIO、定时器、EXTI、DMA、ADC、USART、SPI和I2C等,并分别给出在KEIL MDK下

采用库函数方式使用这些片上外设 / 接口进行应用开发的典型案例。

与传统的嵌入式系统书籍相比,本书具有以下特点:

第一,从读者认知的角度出发,以嵌入式系统的组成为线索,采用自下而上的方法,从硬件到软件依次介绍嵌入式系统;内容的组织、安排合理,结构清晰,系统性强,易于阅读和理解。

第二,将嵌入式系统的一般原理与实际应用开发相结合。本书从“共性”到“个性”,从一般到具体,以嵌入式系统的一般原理为主线,结合目前主流的嵌入式处理器——基于 ARM Cortex-M3 内核的微控制器 STM32F103 和常用的嵌入式集成开发工具——KEIL MDK,由内核而外设逐步讲解,并给出典型应用实例,使读者掌握嵌入式系统基本理论知识的同时,具备一定的 STM32F103 微控制器应用程序的开发和调试能力。

第三,重视硬件,强调底层。目前,大多数嵌入式系统书籍主要讲述基于嵌入式操作系统上的应用软件开发,而对嵌入式硬件原理和接口设计涉及甚少。因此,本书考虑到电气信息类专业的知识架构,突出硬件原理讲解,强调底层驱动设计,从最基本的硬件原理和底层硬件出发,讲述无操作系统下的 STM32F103 微控制器开发,期望改变嵌入式系统教学过程中重理论轻实践、软件强硬件弱的现状。

本书得到了上海工程技术大学教材建设项目的资助,其中,第 1、2、4、6~12 章由王益涵编写,第 3 章由孙宪坤编写,第 5 章由史志才编写。本书的撰写得到了作者家人的理解和大力支持,并且一直得到清华大学出版社焦虹老师的关心和大力支持,清华大学出版社的工作人员也付出了辛勤的劳动,在此谨向支持和关心本书编著的作者家人、同仁和朋友一并致谢。

由于嵌入式技术发展日新月异,加之作者水平有限及时间仓促,书中难免有差错和不足之处,恳请广大读者批评指正。如果读者对本书有任何建议、意见和想法,请与作者联系,作者的电子邮箱:chan@sues.edu.cn。

王益涵

2016 年 3 月

第1篇 系统篇

第1章 嵌入式系统概述	3
1.1 嵌入式系统的定义和特点	3
1.1.1 嵌入式系统的定义	3
1.1.2 嵌入式系统和通用计算机比较	4
1.1.3 嵌入式系统的特点	5
1.2 嵌入式系统的硬件	6
1.2.1 嵌入式处理器	7
1.2.2 嵌入式存储器	15
1.2.3 嵌入式 I/O 设备	18
1.2.4 嵌入式 I/O 接口	18
1.3 嵌入式系统的软件	21
1.3.1 无操作系统的嵌入式软件	22
1.3.2 带操作系统的嵌入式软件	24
1.4 嵌入式系统的分类	27
1.4.1 按硬件(嵌入式处理器)划分	27
1.4.2 按软件复杂度划分	27
1.4.3 按实时性划分	28
1.4.4 按使用对象划分	28
1.5 嵌入式系统的应用	28
1.5.1 国防军事	28
1.5.2 工业控制	29
1.5.3 消费电子	30
1.5.4 办公自动化产品	30
1.5.5 网络和通信设备	30
1.5.6 汽车电子	31



1.5.7	金融商业	31
1.5.8	生物医学	32
1.5.9	信息家电	32
1.6	本章小结	34
	习题 1	34
第 2 章	嵌入式系统开发	35
2.1	嵌入式系统的开发环境、开发工具和调试方式	35
2.1.1	嵌入式系统的开发环境	35
2.1.2	嵌入式系统的开发工具	37
2.1.3	嵌入式系统的调试方式	43
2.2	嵌入式系统的开发语言	50
2.2.1	嵌入式硬件开发语言	50
2.2.2	嵌入式软件开发语言	51
2.3	嵌入式系统的开发过程	53
2.3.1	需求分析	54
2.3.2	系统设计	55
2.3.3	系统实现	61
2.3.4	系统测试	70
2.3.5	系统发布	73
2.4	嵌入式开发工程师之路	74
2.4.1	嵌入式行业和人才的现状分析	74
2.4.2	嵌入式开发工程师的能力要求	74
2.4.3	嵌入式开发工程师的进阶之路	75
2.5	本章小结	77
	习题 2	78

第 2 篇 内 核 篇

第 3 章	ARM Cortex-M3 处理器	81
3.1	ARM Cortex-M3 组成结构	81
3.1.1	Cortex-M3 内核	82
3.1.2	调试系统	84
3.2	ARM Cortex-M3 总线接口	86
3.2.1	Cortex-M3 总线接口类型	87
3.2.2	Cortex-M3 总线连接方案	88
3.3	ARM Cortex-M3 编程模型	89

3.3.1	工作状态	89
3.3.2	数据类型	89
3.3.3	寄存器	89
3.3.4	指令系统	93
3.3.5	操作模式和特权分级	96
3.3.6	异常和中断	98
3.3.7	双堆栈机制	105
3.4	ARM Cortex-M3 存储器系统	107
3.4.1	存储器映射	107
3.4.2	位带操作	110
3.4.3	存储格式	112
3.5	ARM Cortex-M3 的低功耗模式	113
3.6	本章小结	114
	习题 3	115
第 4 章	基于 ARM Cortex-M3 的 STM32 微控制器	117
4.1	从 Cortex-M3 到基于 Cortex-M3 的 MCU	117
4.2	基于 Cortex-M3 的 STM32 系列微控制器概述	118
4.2.1	产品线	118
4.2.2	命名规则	124
4.2.3	生态系统	125
4.2.4	开发方法	131
4.2.5	学习之路	134
4.3	STM32F103 微控制器基础	136
4.3.1	概述	136
4.3.2	主系统结构	137
4.3.3	功能模块	139
4.3.4	引脚定义	140
4.3.5	存储器组织	141
4.4	STM32F103 微控制器的最小系统	145
4.4.1	电源电路	145
4.4.2	时钟电路	148
4.4.3	复位电路	149
4.4.4	调试和下载电路	150
4.4.5	其他	151
4.5	STM32F103 微控制器的时钟系统	153
4.5.1	输入时钟	153
4.5.2	系统时钟	155



4.5.3	由系统时钟分频得到的其他时钟	155
4.5.4	STM32F10x 时钟系统相关库函数	157
4.6	STM32F103 微控制器的低功耗模式	162
4.6.1	睡眠模式	163
4.6.2	停机模式	163
4.6.3	待机模式	164
4.7	STM32F103 微控制器的安全特性	165
4.7.1	看门狗	165
4.7.2	电源检测	166
4.7.3	时钟安全系统	166
4.8	STM32F103 微控制器的启动过程	167
4.8.1	启动过程和启动代码概述	167
4.8.2	ARM 启动代码所需汇编语言基础	167
4.8.3	STM32F103 的启动代码分析	171
4.8.4	STM32F103 的启动过程分析	182
4.9	建立第一个 STM32F103 应用工程	185
4.9.1	STM32F10x 标准外设库的下载和认知	185
4.9.2	嵌入式开发工具的下载和安装	198
4.9.3	配置 STM32F103 工程	198
4.9.4	编写用户程序源代码	206
4.9.5	编译和链接 STM32F103 工程	207
4.9.6	调试 STM32F103 工程	208
4.9.7	将可执行程序下载到 STM32F103 运行	215
4.10	本章小结	217
	习题 4	217

第 3 篇 片内外设篇

第 5 章	GPIO	221
5.1	GPIO 概述	221
5.2	STM32F103 的 GPIO 工作原理	221
5.2.1	内部结构	222
5.2.2	工作模式	224
5.2.3	输出速度	224
5.2.4	复用功能重映射	225
5.2.5	外部中断映射和事件输出	228
5.2.6	主要特性	228
5.3	STM32F10x 的 GPIO 相关库函数	228

5.3.1	GPIO_DeInit	229
5.3.2	GPIO_Init	229
5.3.3	GPIO_SetBits	231
5.3.4	GPIO_ResetBits	231
5.3.5	GPIO_Write	232
5.3.6	GPIO_ReadOutputDataBit	232
5.3.7	GPIO_ReadOutputData	232
5.3.8	GPIO_ReadInputDataBit	233
5.3.9	GPIO_ReadInputData	233
5.3.10	GPIO_EXTILineConfig	233
5.3.11	GPIO_PinRemapConfig	234
5.4	STM32F103 的 GPIO 开发实例——LED 闪烁	235
5.4.1	功能要求	235
5.4.2	硬件设计	235
5.4.3	软件流程设计	236
5.4.4	软件代码实现	236
5.4.5	软件模拟仿真	238
5.4.6	下载到硬件运行	241
5.4.7	开发经验小结——STM32 微控制器开发的一般步骤	241
5.5	STM32F103 的 GPIO 开发实例——按键控制 LED 亮灭	241
5.5.1	功能要求	241
5.5.2	硬件设计	241
5.5.3	软件流程设计	242
5.5.4	软件代码实现	243
5.5.5	软件模拟仿真	244
5.5.6	下载到硬件运行	246
5.5.7	开发经验小结——使用库函数开发 STM32F103 的 GPIO	246
5.6	本章小结	247
	习题 5	248

第 6 章 定时器

6.1	定时器概述	249
6.1.1	延时的实现	249
6.1.2	可编程定时/计数器功能概述	251
6.2	STM32F103 的定时器概述	252
6.3	STM32F103 的基本定时器 TIM6 和 TIM7	252
6.3.1	内部结构	253
6.3.2	时钟源	253



6.3.3	计数模式	254
6.3.4	主要特性	254
6.4	STM32F103 的通用定时器 TIM2—TIM5	254
6.4.1	内部结构	254
6.4.2	时钟源	256
6.4.3	计数模式	257
6.4.4	输出比较模式	259
6.4.5	PWM 输出模式	259
6.4.6	输入捕获模式	262
6.4.7	PWM 输入模式	262
6.4.8	单脉冲模式	263
6.4.9	编码器接口	263
6.4.10	主要特性	263
6.5	STM32F103 的高级定时器 TIM1 和 TIM8	264
6.5.1	内部结构	264
6.5.2	时钟源	264
6.5.3	功能描述	266
6.5.4	主要特性	266
6.6	STM32F10x 定时器相关库函数	266
6.6.1	TIM_DeInit	267
6.6.2	TIM_TimeBaseInit	268
6.6.3	TIM_OC1Init	269
6.6.4	TIM_OC2Init	270
6.6.5	TIM_OC3Init	271
6.6.6	TIM_OC4Init	271
6.6.7	TIM_OC1PreloadConfig	272
6.6.8	TIM_OC2PreloadConfig	272
6.6.9	TIM_OC3PreloadConfig	272
6.6.10	TIM_OC4PreloadConfig	273
6.6.11	TIM_ARRPreloadConfig	273
6.6.12	TIM_CtrlPWMOutputs	274
6.6.13	TIM_Cmd	274
6.6.14	TIM_GetFlagStatus	275
6.6.15	TIM_ClearFlag	275
6.6.16	TIM_ITConfig	276
6.6.17	TIM_GetITStatus	276
6.6.18	TIM_ClearITPendingBit	277
6.7	STM32F103 定时器开发实例——精确定时的 LED 闪烁	277

6.7.1	功能要求	277
6.7.2	硬件设计	277
6.7.3	软件流程设计	278
6.7.4	软件代码实现	279
6.7.5	软件模拟仿真	281
6.7.6	下载到硬件运行	283
6.7.7	开发经验小结——使用 printf 在调试窗口输出	284
6.8	STM32F103 定时器开发实例——PWM 输出	285
6.8.1	功能要求	285
6.8.2	硬件设计	286
6.8.3	软件流程设计	286
6.8.4	软件代码实现	288
6.8.5	软件模拟仿真	289
6.8.6	下载到硬件运行	291
6.8.7	开发经验小结——基于无限循环的嵌入式软件架构	291
6.9	本章小结	292
	习题 6	292
第 7 章	中断	293
7.1	中断的基本概念	293
7.1.1	中断源	294
7.1.2	中断屏蔽	294
7.1.3	中断处理过程	294
7.1.4	中断优先级	296
7.1.5	中断嵌套	297
7.1.6	中断的利与弊	298
7.2	STM32F103 中断系统	298
7.2.1	嵌套向量中断控制器 NVIC	298
7.2.2	STM32F103 中断优先级	299
7.2.3	STM32F103 中断向量表	300
7.2.4	STM32F103 中断服务函数	303
7.2.5	STM32F103 中断设置过程	304
7.3	STM32F103 外部中断/事件控制器 EXTI	306
7.3.1	内部结构	307
7.3.2	工作原理	308
7.3.3	主要特性	309
7.4	STM32F10x 的 NVIC 相关库函数	309
7.4.1	NVIC_PriorityGroupConfig	310

7.4.2	NVIC_Init	310
7.4.3	NVIC_DeInit	313
7.5	STM32F10x 的 EXTI 相关库函数	314
7.5.1	EXTI_DeInit	314
7.5.2	EXTI_Init	314
7.5.3	EXTI_GetFlagStatus	316
7.5.4	EXTI_ClearFlag	316
7.5.5	EXTI_GetITStatus	317
7.5.6	EXTI_ClearITPendingBit	317
7.6	STM32F103 的中断开发实例——按键控制 LED 亮灭	318
7.6.1	功能要求	318
7.6.2	硬件设计	318
7.6.3	软件流程设计	319
7.6.4	软件代码实现	320
7.6.5	下载到硬件运行	323
7.6.6	开发经验小结——前/后台嵌入式软件架构	323
7.7	STM32F103 的中断开发实例——精确延时的 LED 闪烁	325
7.7.1	功能要求	325
7.7.2	硬件设计	325
7.7.3	软件流程设计	326
7.7.4	软件代码实现	328
7.7.5	软件代码分析——volatile	331
7.7.6	软件模拟仿真	332
7.7.7	下载到硬件运行	334
7.7.8	开发经验小结——改进的前/后台嵌入式软件架构	334
7.8	本章小结	336
	习题 7	336
第 8 章	DMA	337
8.1	DMA 的基本概念	337
8.1.1	DMA 的引入	337
8.1.2	DMA 的定义	338
8.1.3	DMA 传输要素	338
8.1.4	DMA 传输过程	338
8.1.5	DMA 的特点与应用	339
8.2	STM32F103 的 DMA 工作原理	339
8.2.1	功能框图	340
8.2.2	触发通道	341

8.2.3	优先级	344
8.2.4	传输模式	344
8.2.5	主要特性	344
8.3	STM32F10x 的 DMA 相关库函数	345
8.3.1	DMA_DeInit	346
8.3.2	DMA_Init	346
8.3.3	DMA_GetCurrDataCounter	348
8.3.4	DMA_Cmd	348
8.3.5	DMA_GetFlagStatus	349
8.3.6	DMA_ClearFlag	350
8.3.7	DMA_ITConfig	351
8.3.8	DMA_GetITStatus	351
8.3.9	DMA_ClearITPendingBit	353
8.4	STM32F103 的 DMA 开发实例——存储器间的数据传输	353
8.4.1	功能要求	353
8.4.2	硬件设计	353
8.4.3	软件流程设计	354
8.4.4	软件代码实现	355
8.4.5	软件代码分析——const	358
8.4.6	下载硬件调试	358
8.4.7	开发经验小结——使用 DMA	365
8.5	本章小结	366
	习题 8	366
第 9 章	ADC	367
9.1	ADC 概述	367
9.1.1	ADC 的由来	367
9.1.2	ADC 的基本原理	368
9.1.3	ADC 的性能参数	370
9.1.4	ADC 的主要类型	371
9.2	STM32F103 的 ADC 工作原理	372
9.2.1	主要特性	373
9.2.2	内部结构	373
9.2.3	ADC 通道及分组	375
9.2.4	ADC 触发转换	376
9.2.5	ADC 时钟和转换时间	377
9.2.6	ADC 工作过程	378
9.2.7	ADC 中断和 DMA 请求	378



9.2.8	独立模式和双 ADC 模式	380
9.2.9	单次和连续转换模式	380
9.2.10	扫描模式	381
9.2.11	间断模式	383
9.2.12	校准	384
9.3	STM32F10x 的 ADC 相关库函数	384
9.3.1	ADC_DeInit	386
9.3.2	ADC_Init	386
9.3.3	ADC_RegularChannelConfig	388
9.3.4	ADC_InjectedChannelConfig	390
9.3.5	ADC_InjectedSequencerLengthConfig	390
9.3.6	ADC_SetInjectedOffset	391
9.3.7	ADC_TampSensorVrefintCmd	391
9.3.8	ADC_Cmd	392
9.3.9	ADC_ResetCalibration	392
9.3.10	ADC_GetResetCalibrationStatus	392
9.3.11	ADC_StartCalibration	393
9.3.12	ADC_GetCalibrationStatus	393
9.3.13	ADC_SoftwareStartConvCmd	394
9.3.14	ADC_ExternalTrigConvCmd	394
9.3.15	ADC_SoftwareStartInjectedConvCmd	395
9.3.16	ADC_ExternalTrigInjectedConvCmd	395
9.3.17	ADC_ExternalTrigInjectedConvConfig	396
9.3.18	ADC_AutoInjectedConvCmd	397
9.3.19	ADC_DiscModeCmd	397
9.3.20	ADC_DiscModeChannelCountConfig	398
9.3.21	ADC_InjectedDiscModeCmd	398
9.3.22	ADC_GetConversionValue	398
9.3.23	ADC_GetInjectedConversionValue	399
9.3.24	ADC_GetFlagStatus	399
9.3.25	ADC_ClearFlag	400
9.3.26	ADC_ITConfig	400
9.3.27	ADC_GetITStatus	401
9.3.28	ADC_ClearITPendingBit	401
9.3.29	ADC_DMACmd	402
9.4	STM32F103 的 ADC 开发实例——读取 GPIO 引脚电压	402
9.4.1	功能要求	402
9.4.2	硬件设计	402

9.4.3	软件流程设计	403
9.4.4	软件代码实现	406
9.4.5	软件模拟仿真	409
9.4.6	下载到硬件调试	413
9.4.7	开发经验小结——使用软件滤波降低噪声	414
9.5	STM32F103 的 ADC 开发实例——读取芯片温度	414
9.5.1	功能要求	414
9.5.2	硬件设计	414
9.5.3	软件流程设计	415
9.5.4	软件代码实现	418
9.5.5	下载到硬件调试	421
9.5.6	开发经验小结——轮询、中断和 DMA	422
9.6	本章小结	423
	习题 9	423
第 10 章	UART	425
10.1	数据通信的基本概念	426
10.1.1	并行和串行	426
10.1.2	单工、半双工和全双工	426
10.1.3	同步和异步	426
10.2	UART 通信原理	427
10.2.1	UART 的物理层	427
10.2.2	UART 的协议层	430
10.3	STM32F103 的 USART 工作原理	432
10.3.1	主要特性	432
10.3.2	内部结构	432
10.3.3	USART 中断	435
10.3.4	使用 DMA 进行 USART 通信	436
10.4	STM32F10x 的 USART 相关库函数	437
10.4.1	USART_DeInit	438
10.4.2	USART_Init	438
10.4.3	USART_Cmd	439
10.4.4	USART_SendData	440
10.4.5	USART_ReceiveData	440
10.4.6	USART_GetFlagStatus	441
10.4.7	USART_ClearFlag	441
10.4.8	USART_ITConfig	442
10.4.9	USART_GetITStatus	443

10.4.10	USART_ClearITPendingBit	444
10.4.11	USART_DMAMCmd	444
10.5	STM32F103 的 USART 开发实例——重定向 printf	445
10.5.1	功能要求	445
10.5.2	硬件设计	445
10.5.3	软件流程设计	446
10.5.4	软件代码实现	447
10.5.5	下载到硬件运行	449
10.5.6	开发经验小结——使用 printf 重定向到 USART1	451
10.6	STM32F103 的 USART 开发实例——PC 串口通信	452
10.6.1	功能要求	452
10.6.2	硬件设计	452
10.6.3	软件流程设计	452
10.6.4	软件代码实现	452
10.6.5	下载到硬件运行	454
10.6.6	开发经验小结——库函数开发 STM32F103 外设的一般原理	455
10.7	本章小结	459
	习题 10	459
第 11 章 SPI		461
11.1	SPI 通信原理	461
11.1.1	SPI 的物理层	462
11.1.2	SPI 的协议层	464
11.2	STM32F103 的 SPI 工作原理	469
11.2.1	主要特性	469
11.2.2	内部结构	470
11.2.3	SPI 主模式	471
11.2.4	SPI 从模式	472
11.2.5	SPI 状态标志和中断	473
11.2.6	SPI 发送数据和接收数据	474
11.2.7	使用 DMA 进行 SPI 通信	475
11.3	STM32F10x 的 SPI 相关库函数	476
11.3.1	SPI_I2S_DeInit	477
11.3.2	SPI_Init	477
11.3.3	SPI_Cmd	479
11.3.4	SPI_I2S_SendData	479
11.3.5	SPI_I2S_ReceiveData	480
11.3.6	SPI_I2S_GetFlagStatus	480