



周立功单片机公司策划

嵌入式实时操作系统

Small RTOS51

原理及应用

Small RTOS51

Small RTOS51

陈明计 周立功 等编著



 北京航空航天大学出版社
<http://www.buaapress.com.cn>

嵌入式实时操作系统 Small RTOS51 原理及应用

陈明升 周立功 等编著

北京航空航天大学出版社

<http://www.buaapress.com.cn>

内 容 简 介

Small RTOS51 是为 51 系列单片机而编写的。它是完全免费的、源代码公开的多任务实时操作系统。它可在无任何外部数据存储器的单片 80C51 系统上运行,并且是可移植的。全书分两部分。第一部分为基础篇,介绍 Small RTOS51 和一些基本概念,并详细分析 Small RTOS51 的工作原理。第二部分为应用篇,给出部分常用硬件在 Small RTOS51 下驱动程序的源代码。这些源代码在 DP-51 单片机仿真实验仪上全部调试通过,且只要经过很少的改动,或是不改动,就可以在其他环境下使用。应用篇还通过对这些源代码的分析,让读者理解基于 RTOS 的编程方法,并给出完整的例子让读者全面掌握基于 RTOS 的编程方法。

本书可以作为高等院校相关专业的课程教材、实验参考资料或课外读物,对嵌入式应用开发人员也有重要的参考价值。

图书在版编目(CIP)数据

嵌入式实时操作系统 Small RTOS51 原理及应用/陈明
计等编著. —北京:北京航空航天大学出版社,2004. 1
ISBN 7-81077-409-3

I. 嵌… II. 陈… III. 实时操作系统, Small RTO
S51 IV. TP316.2

中国版本图书馆 CIP 数据核字(2003)第 095727 号

嵌入式实时操作系统 Small RTOS51 原理及应用

陈明计 周立功 等编著

责任编辑 刘晓明

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:(010)82317024 传真:(010)82328026

http://www.buaapress.com.cn E-mail:bhpress@263.net

河北省涿州市新华印刷厂印制 各地书店经销

*

开本:787×1 092 1/16 印张:22.5 字数:576 千字

2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷 印数:4 000 册

ISBN 7-81077-409-3

定价:38.00 元(含光盘)

快速掌握嵌入式操作系统的捷径

2002 年的某一天下午,有位员工告诉我,他发现有人在“21IC 中国电子网”发布了一个基于 80C51 单片机、仅需要 4 KB 程序空间和 128 个 RAM 单元就可以运行的嵌入式操作系统自由软件源码。我不由自主地想到了这个人可能就是公司要寻找的优秀人才,于是开始静下心来花了一个通宵大致阅读了整个源码,发现作者编写的程序具有非常良好的风格。凭我多年的经验判断,作者是一个有强烈爱好和敬业精神的人,于是开始给作者写 E-mail 联系。当然,现在他已经成为了我们公司的员工,他就是本书的主要作者陈明计先生。他目前主要负责嵌入式操作系统应用部门的工作,领导一个小组从事 PHILIPS 公司半导体 32 位 ARM 单片机的应用开发。

去年,我邀请清华大学邵贝贝教授来公司专门讲解 $\mu\text{C}/\text{OS}-\text{II}$ 嵌入式操作系统。为了帮助内部开发工程师深入掌握嵌入式操作系统的应用技术,我要求公司的开发工程师尝试写作一个基于 80C51 单片机的小操作系统。有一部分员工完成了这个大作业,可以说效果十分惊人,完全达到了理解和掌握嵌入式操作系统的目的。由此可以看出,这是一种行之有效的学习方法。

面对未来,一个要继续从事电子产品开发的工程师,不懂得嵌入式操作系统是不可想像的。怎么办?实践证明,最好的办法就是尝试自己写一个简易的嵌入式操作系统。不在乎写出来的嵌入式操作系统水平的高低,关键是通过做这样一件事情,将完全有助于彻底弄懂嵌入式操作系统的来龙去脉。当然,对于大多数人来说,要自己去写一个嵌入式操作系统软件不是一件容易的事情;但如果能够真正读懂一个嵌入式操作系统的源码,从而达到运用自如地编写驱动程序“中间件”的程度,也不失为一条捷径。我认为,经过大量实际应用,使用 Keil C51 编写的嵌入式实时操作系统 Small RTOS51 就是一个很好的例子,特此推荐给广大的单片机应用开发工程师。它将为您迈入 32 位嵌入式操作系统的应用开发行列打下一个坚实的基础。

QJSY/c

基于此,由我负责组织和策划并调动了公司相关开发小组的优秀人才和内部的技术资源,成立了一个专门的小组来写作这本专著。其中围绕 DP-51 单片机仿真实验仪的硬件电路阐述 Small RTOS51 操作系统的设计原理,详细地介绍了 RS232、I²C、USB 1.1 和 CANBUS 等一系列外围器件及其功能部件的应用设计。我们有针对性地提供了大量的源码,希望能够最大限度地帮助更多的初学者掌握嵌入式操作系统,跟上新技术发展的潮流。

本书的第一部分为原理篇,主要介绍 Small RTOS51 的设计思想和实现方法,

读者可以结合 Small RTOS51 源程序和本篇的详细解说,深入研究其中的机理。第二部分为应用篇,主要介绍 Small RTOS51 的基本应用,以实战为主线,详细地阐述了单片机基本功能部件的应用及其外围器件的应用设计。后续的内容主要介绍基于 Small RTOS51 的 USB 固件编程及其 CANBUS 的驱动开发。这部分有一定的难度,牵涉的知识面比较广,对开发工程师本身的要求很高,读者可以根据自己的具体情况作为选学内容。

参与本书编写及其相关工作的主要人员有陈明计、刘养海、刘英斌、黄晓清、黄绍斌;全书由周立功负责规划、审核与统稿。很多的网友多次来信讨论对编写这本书的意见,也进一步征求了不少在教学中引入 80C51 操作系统的老师的意见,在此对他们表示衷心的感谢。有一点非常重要,如果没有陈明计个人的努力,也就没有 Small RTOS51 的诞生,无疑这是不能忘记的!

特别要感谢北航出版社的全力支持,如果没有他们的努力和辛勤劳动,这本书不会这么快出版。

周立功

2003 年 8 月 28 日

前 言

在激烈的竞争下,公司(特别是小公司)往往因为市场的压力,要求开发人员在较短的时间内把产品开发出来,从而导致产品不稳定和维护困难。我使用 8051 系列单片机,在当时是用汇编语言编写程序。我于 1998 年开始接触 C51 高级语言,从 1999 年开始直到现在,一直使用 C 语言编写 8051 的程序。在此期间,我经常思考如何在保证性能的前提下提高开发效率,并降低程序的维护难度。根据我的经验,使用 C 语言的确会提高开发效率,也能提高软件的可维护性;但仅仅依靠编程语言还远远不够,应该要建立自己的开发平台。在平台上开发,才能事半功倍。嵌入式实时操作系统(RTOS)就是一个很好的嵌入式软件的开发平台。近年来,随着半导体工业的发展,32 位单片机的价格持续降低。在某些应用中,使用 32 位单片机的总体成本更低,而在 32 位单片机中不使用 RTOS 几乎是不可能的。因而,在嵌入式系统中使用 RTOS 是大势所趋。

到 2001 年,我下决心学习 RTOS。当时我仍旧是用 8051 系列单片机开发产品,于是想通过一个基于 51 单片机的 RTOS 来学习它。当时基于 51 单片机的 RTOS 中比较有名气的有 Keil C51 所带的 RTX Full 和 RTX Tiny 及网友移植的 $\mu\text{C}/\text{OS}-\text{II}$ 。以下对这几个 RTOS 进行简单的介绍。

RTX51 是一个用于 8051 系列单片机的多任务实时操作系统。有两个不同的 RTX51 版本可以利用。其中 RTX51 Full 使用四个任务优先权完成同时存在时间片轮转调度和抢先的任务切换。RTX51 工作在与中断功能相似的状态下,信号和信息可以通过邮箱系统在任务之间互相传递。您可以从一存储池中分配和释放内存;可以强迫一个任务等待中断、超时,或者是从另一个任务或中断发出信号、信息。而 RTX51 Tiny 是一个 RTX51 的子集,可以很容易地在没有任何外部存储器的单片 8051 系统上运转;但它仅支持时间片轮转任务切换和使用信号进行任务切换(即非抢占式的),不支持抢占式的任务切换,不包括消息队列,没有存储器池分配程序。

$\mu\text{C}/\text{OS}-\text{II}$ 是著名的、源码公开的实时内核,可用于各类 8 位、16 位和 32 位单片机或 DSP。从 $\mu\text{C}/\text{OS}$ 算起,该内核已有 10 余年应用史,并在诸多领域得到广泛应用。 $\mu\text{C}/\text{OS}-\text{II}$ 是一个完整、可移植、可固化和可剪裁的抢先式实时多任务内核。 $\mu\text{C}/\text{OS}-\text{II}$ 是用 ANSI 的 C 语言编写的,包含一小部分汇编代码,使之可以供不同架构的微处理器使用。至今,从 8 位到 64 位, $\mu\text{C}/\text{OS}-\text{II}$ 已在超过 40 种不同架构的微处理器上运行。

8051 系列单片机一般只有很少的 ROM 和 RAM 资源,如 P89C51 只有 4 KB

Flash 和 128 字节 RAM。但 RTX51 Full 自身代码有 6 K 多字节,且需要大量外部 RAM,又无源代码,很多时候不实用,不利于学习。RTX Tiny 虽然小(自身占用 900 多字节 ROM),但是任务没有优先级和中断管理,也无源代码,也不太实用(目前 Keil 已经把 RTX Tiny 的源码提供给它正版用户,全部是汇编代码),也不利于学习。而 μ C/OS-II 有源代码,也有配套图书,比较利于学习。但是它规模太大,又需要大量外部 RAM,而且所有函数都必须是可重入函数,用在 8051 这类小片内 RAM 的单片机上有点勉强,在 8051 系列单片机上不太实用。

于是为了学习 RTOS,也为了建立自己的开发平台,到 2002 年春节,我下决心自己编写一个基于 8051 系列单片机的 RTOS,这就是 Small RTOS51 V 0.50 版。它使用了 RTX51 Tiny 的堆栈管理机制,并像 μ C/OS-II 一样是抢占式的。虽然它为 51 系列单片机编写,但比较容易移植到其他 CPU 上。

到 2002 年 6 月,经过一些实践的检验,Small RTOS51 升级到 V 1.00 版,这也是公开发布的第一个版本。目前 Small RTOS51 最新版本为 V 1.12.1 版,V 1.20.0 版本也已基本完成,在本书发售时应该已经发布,读者可以到 <http://www.zlgmcu.com/philips/philips-embedsys.asp> 下载最新版。目前 Small RTOS51 的所有版本均可以免费在任何领域使用,详细情况请参考使用许可证(见附录 A)。V 1.20.0 版与 V 1.12.1 版相比,主要有以下改变:

支持任务动态建立与删除,函数功能向一般的 RTOS 靠拢;支持 C51 的重入函数(用关键字 reentrant 定义的函数);支持动态内存分配(使用动态内存分配的任务必须使用重入栈),并增加一个对 PHILIPS 公司的 ARM(LPC2106/LPC2105/LPC2104)的移植。

本书的原理篇(第 0 章到第 10 章)以 Small RTOS51 V 1.12.1 版为基础,通过分析源码介绍 Small RTOS51 原理。第 0 章源自与 Small RTOS51 V 1.00 版本同时发布的我撰写的文章《单片机 RTOS 随想曲》,即现在的绪论,主要说明学习嵌入式实时操作系统的必要性。第 1 章简单介绍 Small RTOS51 的特点、运行条件等知识。第 2 章通过一个简单的例子说明如何使用 Small RTOS51。第 3 章介绍一些基本概念。第 4 章分析任务管理的代码。这部分是 RTOS 的真正核心。核心的其他部分,如消息队列、延时/超时等都可以看作是基于 RTOS 的中间件(或者说驱动程序)。这一章还分析了延时/超时处理代码。第 5 章则详细分析了 Small RTOS51 与 CPU 相关的代码,揭示了 8051 系列单片机如何从一个任务切换到另一个任务。对于初学者,这是 RTOS 中比较神秘的地方。第 6 章和第 7 章介绍 Small RTOS51 的两种通信方式:信号量和消息队列。它们与标准的信号量和消息队列有细微差别。第 8 章介绍如何把 Small RTOS51 移植到其他的微处理器框架上。第 9 章和第 10 章分别给出函数参考手册和系统配置手册,以使用户使用 Small RTOS51。

本书的应用篇(第 11 章到第 24 章)以 DP-51 单片机仿真实验仪为硬件基础,介绍各种常用硬件 Small RTOS51 下的中间件(驱动程序)的编写方法,并在最后给出一个完整的实例,说明如何编写基于 RTOS 的程序。这些中间件虽然基于 DP-51 单片机仿真实验仪,但是其与硬件相关的部分基本独立,很容易移植到其他硬件上。第 11 章和第 12 章介绍 DP-51 单片机仿真实验仪的硬件,以使用户容易读懂后面的程序。第 13 章简单介绍如何为 Small RTOS51 编写中间件(驱动程序),这对使用其他 RTOS 也有一定的参考价值。从第 14 章到第 23 章分别分析了各种常用硬件(LCD 扫描、LCD 显示、软定时器、键盘扫描、串行通信、I²C 总线、串行 E²PROM、串行实时时钟、CAN 总线和 USB)的驱动程序(中间件)的编写方法。它们都是采用首先介绍原理;然后以分析源代码的形式编写其驱动程序的写作方法,一般还给出应用实例。第 24 章给出一个比较完整的例子,说明如何编写基于 RTOS 的程序。

对于本书的出版,首先要感谢使用和关心 Small RTOS51 的网友,没有他们,Small RTOS51 也许是昙花一现;更重要的是要感谢我的家人。Small RTOS51 及本书基本上是利用业余时间编写的,我的家人使我有充足的时间完成 Small RTOS51 及本书。但最重要的是要感谢周立功先生及其公司的员工。本书实际上是由周立功先生负责策划的,并给本书的编写提供了很多指导和建议,而且还引用了很多广州周立功单片机技术有限公司的内部资料作为参考。参与本书编写的还有刘养海、刘英斌、黄晓清、黄绍斌。

由于我是第一次写书,且对嵌入式实时操作系统的理解并不深刻,若书中有不恰当甚至完全错误的地方,恳请各位同行指正。

陈明计

2003 年 8 月 28 日

广州周立功单片机发展有限公司

地址:广州市天河区天河北路 689 号光大银行大厦 16 楼 D2

邮编:510630

电话:(020) 38730916 38730917 38730976 38730977 传真:(020)38730925

电子邮箱:info@zlgmcu.com 网址:http://www.zlgmcu.com

技术支持

电话:(020)85520995 85539796 85547386 85541621 85541773

电子邮箱:ARM@zlgmcu.com

广州周立功专卖店

地址:广州市天河区新赛格电子城 203~204 室

邮编:510630

电话:(020)87578634 传真:(020)87578842

深圳周立功

地址:深圳市深南中路 2070 号电子科技大厦 C 座 18 楼 D2

邮编:518031

电话:(0755)83287588 传真:(0755)83274266

成都周立功

地址:成都市一环路南一段 57 号金城大厦 618 室

邮编:610041

电话:(028)85499320 传真:(028)85439505

北京周立功

地址:北京市海淀区黄庄知春路 113 号银网中心 715 室(中发电子市场斜对面)

邮编:100086

电话:(010)62536178 62536179 82614433 传真:(010)82614433

上海周立功

地址:上海市中山西路 827 号东华大学上海喜天游大酒店 4 楼 C31 室

邮编:200051

电话:(021)62199015 传真:(021)62199016

杭州周立功

地址:杭州市教工路 2 号杭州电子市场仪器仪表城二楼 555 室

邮编:310012

电话:(0571)88271834 传真:(0571)88271326

南京周立功

地址:南京市珠江路 280 号珠江大厦 2006 室

邮编:210018

电话:(025)3613221 传真:(025)3641103

目 录

第一部分 原理篇

第 0 章 绪 论	(3)
第 1 章 Small RTOS51 简介	(9)
1.1 Small RTOS51 的特点	(9)
1.2 Small RTOS51 的运行条件	(9)
1.3 Small RTOS51 的存储器需求	(9)
1.4 Small RTOS51 的任务堆栈的计算	(10)
1.5 关于可重入功能	(10)
1.6 关于 C51 的库函数	(10)
1.7 关于 51 系列单片机派生类型的多数据指针和数学单元	(10)
1.8 关于 51 系列单片机的寄存器段	(11)
1.9 关于局部变量	(11)
第 2 章 基本概念	(12)
2.1 嵌入式系统	(12)
2.2 实时系统	(12)
2.3 前、后台系统	(13)
2.4 操作系统	(13)
2.5 实时操作系统	(14)
2.6 代码的临界区	(15)
2.7 资 源	(15)
2.8 共享资源	(15)
2.9 任 务	(15)
2.10 任务切换	(15)
2.11 内 核	(16)
2.12 调 度	(16)
2.13 非占先式内核	(16)
2.14 占先式内核	(17)
2.15 可重入性	(17)
2.16 任务优先级	(19)

2.17	信号量	(20)
2.18	死 锁	(20)
2.19	消息队列	(21)
2.20	中 断	(21)
2.21	时钟节拍	(22)
第 3 章	一个简单的例子	(23)
3.1	Small RTOS51 的安装及目录结构	(23)
3.2	例子简介	(23)
3.3	Config.h、Os_cfg.h 和 Os_cpu.h	(24)
3.4	与编译器无关的数据类型	(24)
3.5	OS_ENTER_CRITICAL() 和 OS_EXIT_CRITICAL()	(25)
3.6	main() 函数	(25)
3.7	建立任务	(26)
3.8	删除任务	(26)
3.9	用户任务	(26)
3.10	软非屏蔽中断	(27)
3.11	中断服务程序的编写	(28)
3.12	Os_cpu.h 可改变的其他内容	(28)
3.13	Small RTOS51 的其他注意事项	(29)
第 4 章	任务管理——核心的核心	(30)
4.1	临界区	(30)
4.2	任 务	(30)
4.3	任务状态	(31)
4.4	与任务相关的数据结构	(32)
4.5	任务调度	(36)
4.6	Small RTOS51 中的中断处理	(39)
4.7	时钟节拍	(43)
4.8	Small RTOS51 初始化和启动	(45)
4.9	建立任务	(46)
4.10	任务堆栈	(47)
4.11	删除任务	(47)
4.12	时间服务及任务的挂起和恢复	(47)
4.13	获取当前 Small RTOS51 的版本号	(52)
4.14	OSInt...() 函数	(52)
第 5 章	如何切换任务	(53)
5.1	CPU 可以执行多个任务的原因	(53)

5.2	CPU 怎样运行才能执行多个任务	(53)
5.3	何时进行任务切换	(54)
5.4	Small RTOS51 任务切换时的程序框图	(54)
5.5	数组 OSTaskStackBottom[] 和 Small RTOS51 的堆栈结构	(55)
5.6	变量 OSFastSwap	(57)
5.7	常量数组 OSMapTbl[]	(58)
5.8	软非屏蔽中断的堆栈 SP2[]	(58)
5.9	中断切换函数 OSIntCtxSw()	(58)
5.10	任务主动放弃 CPU——OS_TASK_SW()	(60)
5.11	堆栈变换函数 C_OSCtxSw()	(62)
5.12	恢复新任务环境 LoadCtx()	(69)
5.13	优先级最低的任务 OSIdle()	(72)
5.14	Small RTOS51 初始化函数 OSStart()	(72)
5.15	系统时钟节拍中断 OSTickISR()	(74)
第 6 章	任务之间的通信和同步之信号量	(78)
6.1	概 述	(78)
6.2	使 Keil C51 函数具有重入性的特殊方法	(78)
6.3	数据结构	(79)
6.4	EN_OS_SEM_CHK 宏及相关代码	(80)
6.5	初始化一个信号量	(80)
6.6	等待一个信号量	(82)
6.7	发送一个信号量	(86)
6.8	无等待地请求一个信号量	(89)
6.9	查询信号量	(90)
第 7 章	任务之间的通信和同步之消息队列	(92)
7.1	概 述	(92)
7.2	数据结构及存储空间的分配	(92)
7.3	消息队列发送消息和取得消息的操作	(93)
7.4	EN_OS_Q_CHK 及相关代码	(95)
7.5	建立一个消息队列	(96)
7.6	等待一个消息队列中的消息	(98)
7.7	向消息队列发送一个消息(FIFO)	(102)
7.8	向消息队列发送一个消息(LIFO)	(106)
7.9	同时发送多个消息	(111)
7.10	无等待地从消息队列中取得消息	(111)
7.11	清空一个消息队列	(113)
7.12	查询一个消息队列的状态	(114)

第 8 章 Small RTOS51 的移植	(116)
8.1 系统相关的宏	(116)
8.2 与编译器无关的变量类型	(117)
8.3 OSStart()	(117)
8.4 OSIntCtxSw()	(118)
8.5 OS_TASK_SW()	(118)
8.6 OSTickISR()	(118)
8.7 OSIdle()	(120)
第 9 章 函数参考手册	(121)
9.1 OS_INT_ENTER()函数	(121)
9.2 OSClearSignal()函数	(121)
9.3 OSIntExit()函数	(123)
9.4 OSIntSendSignal()函数	(124)
9.5 OSQAccept()函数	(124)
9.6 OSQCreate()函数	(125)
9.7 OSQFlush()函数	(126)
9.8 OSQIntPost()函数	(127)
9.9 OSQIntPostFront()函数	(128)
9.10 OSQNMsgs()函数	(129)
9.11 OSQPend()函数	(129)
9.12 OSQPost()函数	(130)
9.13 OSQPostFront()函数	(131)
9.14 OSQSize()函数	(132)
9.15 OSRunningTaskID()函数	(133)
9.16 OSSemAccept()函数	(133)
9.17 OSSemCreate()函数	(134)
9.18 OSSemIntPost()函数	(135)
9.19 OSSemPend()函数	(136)
9.20 OSSemPost()函数	(136)
9.21 OSSemQuery()函数	(137)
9.22 OSSendSignal()函数	(138)
9.23 OSStart()函数	(138)
9.24 OSTimeTick()函数	(139)
9.25 OSVersion()函数	(140)
9.26 OSWait()函数	(141)

第 10 章 系统配置手册	(142)
10.1 基本配置	(142)
10.2 消息队列的配置	(143)
10.3 信号量的配置	(144)
10.4 Small RTOS51 的特殊配置	(144)

第二部分 应用篇

第 11 章 DP-51 单片机仿真实验仪简介	(149)
11.1 概 述	(149)
11.2 产品实物	(149)
11.3 功能特点	(150)
11.4 主控 CPU 简介	(150)
11.5 DP-51 单片机仿真实验仪的应用领域	(152)
第 12 章 DP-51 单片机仿真实验仪硬件结构	(153)
12.1 主要器件	(153)
12.2 电路外观	(153)
12.3 应用接口	(155)
12.4 I/O 地址分配	(159)
12.5 跳线选择器	(160)
第 13 章 如何为 Small RTOS51 编写驱动程序	(163)
13.1 Keil C51 简介	(163)
13.2 Keil C51 的特殊性	(164)
13.3 判断 Keil C51 是否可重入的方法	(165)
13.4 使用任务编写驱动程序	(165)
13.5 使用消息队列编写驱动程序	(165)
13.6 使用信号量编写驱动程序	(165)
13.7 复合方法	(166)
第 14 章 LED 扫描程序的实现	(167)
14.1 LED 简介	(167)
14.2 LED 数码显示器简介	(169)
14.3 LED 驱动程序的实现	(172)

第 15 章 LCD 显示驱动的实现	(177)
15.1 点阵字符型 LCD - TC1602A 简介	(177)
15.2 TC1602A 液晶显示器与 DP - 51 单片机仿真实验仪的连接	(181)
15.3 驱动程序的使用	(181)
15.4 对 TC1602A 操作的基本函数	(183)
15.5 初始化 TC1602A 液晶显示器	(184)
15.6 清除指定行	(185)
15.7 在指定位置显示字符串	(186)
15.8 在指定地址向液晶模块写多个字符	(188)
15.9 驱动程序在 DP - 51 单片机仿真实验仪上使用的例子	(189)
第 16 章 软定时器的实现	(192)
16.1 软定时器简介	(192)
16.2 软定时器模块的使用	(192)
16.3 软定时器的数据结构	(193)
16.4 软定时器模块的接口函数	(193)
16.5 软定时器模块的管理	(196)
16.6 软定时器模块在 DP - 51 运用的例子	(200)
第 17 章 键盘扫描程序的实现	(203)
17.1 键盘简介	(203)
17.2 软件去抖动处理	(204)
17.3 连击的处理	(204)
17.4 复合键	(204)
17.5 串 键	(205)
17.6 键盘驱动的实现	(205)
17.7 键盘驱动在 DP - 51 上运用的例子	(211)
第 18 章 串行通信程序的实现	(216)
18.1 概 述	(216)
18.2 串行通信硬件电路	(216)
18.3 通信协议	(217)
18.4 本例使用的串行通信协议	(217)
18.5 本例执行的效果	(217)
18.6 串行通信驱动程序的发送部分	(217)
18.7 串行通信驱动程序的接收部分	(220)
18.8 例子中其他未介绍部分代码	(222)

第 19 章 I²C 总线驱动程序的实现	(224)
19.1 标准模式 I ² C 总线规范简介	(224)
19.2 位传输	(224)
19.3 数据传输	(225)
19.4 仲裁与时钟发生	(226)
19.5 传输协议	(226)
19.6 I ² C 总线驱动程序的实现	(228)
第 20 章 串行 E²PROM 芯片 Cat24WCxx 驱动程序的实现	(238)
20.1 概 述	(238)
20.2 引脚描述	(238)
20.3 串行 E ² PROM 芯片的寻址	(239)
20.4 写操作方式	(241)
20.5 读操作方式	(242)
20.6 Cat24WC02 在 DP-51 中的连接	(244)
20.7 Cat24WCxx 驱动程序的使用	(244)
20.8 Cat24WCxx 的写操作	(245)
20.9 Cat24WCxx 的读操作	(246)
20.10 DP-51 上对 Cat24WC02 读/写操作的例子	(247)
第 21 章 实时时钟芯片 PCF8563 驱动程序的实现	(250)
21.1 实时时钟 PCF8563 简介	(250)
21.2 PCF8563 的基本原理	(251)
21.3 PCF8563 在 DP-51 中的连接	(257)
21.4 PCF8563 驱动程序的使用	(257)
21.5 PCF8563 的写操作	(258)
21.6 PCF8563 的读操作	(259)
21.7 DP-51 上对 PCF8563 读/写操作的例子	(259)
第 22 章 CAN 总线驱动程序的实现	(262)
22.1 CAN 简介	(262)
22.2 CAN 控制器 SJA1000 简介	(262)
22.3 SJA1000 的硬件连接	(264)
22.4 对 SJA1000 的软件控制	(265)
22.5 SJA1000 的 BasicCAN 方式	(265)
22.6 寄存器位功能介绍	(268)
22.7 SJA1000 的编程流程	(278)
22.8 驱动程序的使用	(279)

22.9	对 SJA1000 操作的基本函数	(281)
22.10	用户可修改、可调整的应用层函数	(290)
22.11	CAN 通信在 DP-51 上运用的例子	(295)
第 23 章	PDIUSB12 USB 驱动程序的实现	(299)
23.1	简介	(299)
23.2	USB 设备的枚举过程	(299)
23.3	USB 标准设备请求	(300)
23.4	程序实现方法	(303)
第 24 章	完整的程序——闹钟的实现	(314)
24.1	简介	(314)
24.2	关键代码	(314)
附录 A	使用许可证	(341)
附录 B	本书配套光盘说明	(342)
参考文献		(343)