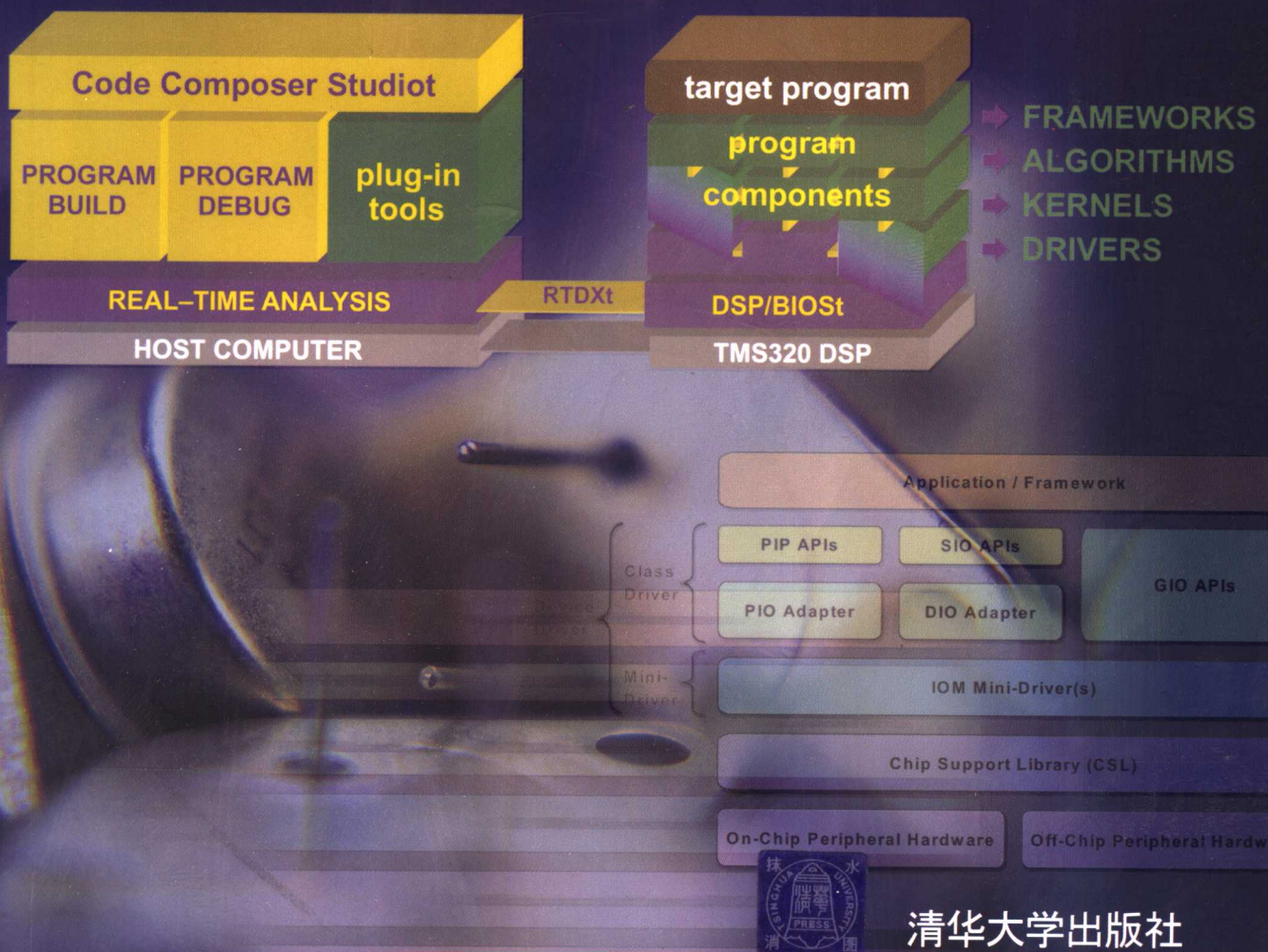


TI DSP系列中文手册

TI DSP/BIOS

用户手册与驱动开发

(美) Texas Instruments Incorporated 著
王军宁 何迪 马娟 胡启龙 等编译



清华大学出版社

TI DSP系列中文手册

TI DSP/BIOS 用户手册与驱动开发

(美) Texas Instruments Incorporated 著

王军宁 何迪 马娟 胡启龙 等编译

清华大学出版社
北京

内 容 简 介

DSP/BIOS 是 TI 公司特别为其 TMS320C6000™, TMS320C5000™ 和 TMS320C28x™ 系列 DSP 平台所设计开发的一个尺寸可裁剪的实时多任务操作系统内核, 是 TI 公司的 Code Composer Studio™ 开发工具的组成部分之一。

本书着重介绍了 DSP/BIOS 用户手册和 DSP/BIOS 设备驱动的开发与使用, 共分 12 章和 4 个附录, 具体内容包括 DSP/BIOS 概述、程序生成、监测、线程调度、存储器和低级函数、I/O 概述和管道、流 I/O 和设备驱动; 设备驱动开发工具包(DDK)简介、DSP/BIOS 设备驱动的结构和使用、使用 DSP/BIOS 设备驱动、GIO 类驱动、微型驱动的开发步骤、IOM 接口、PIO 适配器、LIO 模型到 IOM 模型的移植和 GIO API 的 ASYNC 扩展。

本书可供基于 TI DSP 的程序开发人员、系统设计工程师等参考, 也可以作为相关专业本科生和研究生的选修课程的参考书。

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 13501256678 13801310933

图书在版编目(CIP)数据

TI DSP/BIOS 用户手册与驱动开发/美国德州仪器著; 王军宁等编译. —北京: 清华大学出版社, 2007. 4

(TI DSP 系列中文手册)

ISBN 978-7-302-14124-2

I. T… II. ①美… ②王… III. 数字信号—信息处理系统 IV. TN911.72

中国版本图书馆 CIP 数据核字(2006)第 134535 号

责任编辑: 王敏稚

责任校对: 白 蕾

责任印制: 王秀菊

出版发行: 清华大学出版社 地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编: 100084

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 装 者: 北京市清华园胶印厂

经 销: 全国新华书店

开 本: 185×260 印 张: 17.25 字 数: 401 千字

版 次: 2007 年 4 月第 1 版 印 次: 2007 年 4 月第 1 次印刷

印 数: 1~3000

定 价: 29.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 022950-01

Copyright Grant Letter

2003-4-28

Texas Instruments (Shanghai) Co., Ltd
11F, Novel Plaza, 128 Nanjing Road West,
Huangpu District, Shanghai 200003, P. R. C.

Mr. Hu Guangshu,

We are in possession of a copy of your book draft titled as appendix provided and [printed] by Tsinghua University Press, a book printer and publisher with a principal office located at Xue Yan Mansion, Tsinghua University, Beijing, 100084, P. R. C. (the "Book"), which contains certain copyrighted information (the "Information") from the Texas Instruments TMS320 DSP series product databook.

As rightful owner of the information, we hereby give you permission to use the Information in the book on a non-exclusive basis provided that you place the following statement on the title page of the book.

"This book contains copyrighted material of Texas Instruments Incorporated, used herein with permission of the copyright owner. Errors introduced in the use or translations of the copyrighted material herein are solely the responsibility of the author or translator and are not the responsibility of Texas Instruments Incorporated. Any further use, modification, redistribution without the express approval of the copyright owner is strictly prohibited. This copyright authorization allows for reproduction only in printed and computer materials of the above-cited standards, on a regional scale and for an unlimited period of time. Should any of the copyrighted information fall under patent protection, this copyright authorization is not to be construed as an authorization to use and/or implement patent information without fulfilling attached obligations."

Please also find enclosed some information regarding TI's copyright and trademark policies, which we would request you to follow during the use of the Information.

Thank you for including TI technology in your teaching and scholarship. I welcome your call or E-mail if I can provide additional assistance.

Best Regards,

Eldon Teng
Director of Market Development
Texas Instruments Asia



TI DSP 系列中文手册编译委员会

(按汉语拼音排序)

主任委员:

胡广书	教授	清华大学
彭启琮	教授	电子科技大学
沈 洁	经理	TI 中国大学计划

委员:

陈 健	教授	上海交通大学
戴逸民	教授	中国科学技术大学
何佩琨	教授	北京理工大学
刘和平	教授	重庆大学
潘亚涛	工程师	TI 中国大学计划
桑恩方	教授	哈尔滨工程大学
王军宁	教授	西安电子科技大学
张旭东	副教授	清华大学
曾 刚	编辑	清华大学出版社

序

经过全体编译老师和编译委员会近一年的努力,《TI TMS320 系列 DSP 中文手册》终于陆续和广大读者见面了。

数字信号处理器(Digital Signal Processing, DSP)是对信号和图像实现实时处理的一类高性能的 CPU。所谓“实时(Real-Time)实现”,是指一个实际的系统能在人们听觉、视觉或按任务要求所允许的时间范围内实现对输入信号进行处理并将其输出。目前, DSP 已广泛应用于通信、家电、航空航天、工业测量、控制、生物医学工程及军事等许许多多需要实时实现的领域。

美国德州仪器(Texas Instruments, TI)公司是全球 DSP 研发和生产的领先者。自 1982 年推出第一块 DSP 芯片以来,到 20 世纪 90 年代中期, TI 先后推出了 C10、C20、C30、C40、C50 及 C80 等 6 代 TMS320 系列的 DSP 产品。紧接着又推出了 C2000 系列、C5000 系列和 C6000 系列三大主流产品,并推出了将 DSP 和 ARM 合为一体的 OMAP 系列。这些产品无论在国外还是在国内都获得了广泛的应用。例如,“TI 中国大学计划”在 2003 年举办的“TI DSP 设计比赛”中,国内高校就有约 90 个队参加,足见 DSP 在我国已经得到普遍的重视。

凡是从事过含有 CPU 的系统(单片机或 DSP)的设计人员都知道,为了顺利地实现设计任务,一本或几本好的手册是必不可少的,其中包括该 CPU 的结构手册、指令和汇编语言手册以及开发手册等。

由于 TI 的 DSP 发展迅速,产品更新快,因此其手册自然也非常多。由于手册需要更新和补充,因此,彼此之间难免会出现重复和种类繁多的现象。使用过 TI DSP 文档的用户都感觉到,其手册在使用上是有相当难度的。另外, TI DSP 文档都是用英文写成,这也给部分工程技术人员带来一定的困难。

鉴于此, TI 中国主管提出委托国内的高校老师对其文档进行编译,并授权清华大学出版社正式出版。在“TI 中国大学计划”的建议下, 2003 年 6 月通过推荐和报名方式成立了编译委员会。

通过认真讨论,编译委员会首先确定了文档编译的原则,然后确定了编译的书目,最后确定了每一本书的编译者。

关于编译的原则,我们提出了如下两点:

(1) 本文档的定位为“手册”。也就是说,每一位文档的编译者应全面了解和掌握所编译书目的所有英文文档,并了解各个文档之间的关系,在保证文档完整的基础上,选择最新的文档,并去除其中的重复内容和已经淘汰的内容。

(2) 要尽可能地按照 TI 英文文档的“本意”来形成中文,以保证手册的准确性。允许作

者按自己的经验有所发挥,以便于难点的理解。

这次编译的书目包含三大部分,一是各个系列的共用部分,如 CCS、DSP/BIOS、算法标准、C 语言编译器及开发工具等各个手册;二是按 C2000、C5000 和 C6000 三大系列分别编译它们的 CPU 结构及指令手册;三是分别编译它们的应用。

编译计划在“TI 中国大学计划”的相关会议上提出后,得到了国内高校许多老师的热情支持,很快便将要编译的书目一一落实。这些老师都有着从事 DSP 教学和科研的丰富经验,正是由于他们的大力支持,才使这一庞大的工作计划能够付诸实施。在此,谨向参加本系列手册编译工作的全体老师表示衷心的感谢!

“TI 中国大学计划”在本系列手册的编译过程中给予了多方面的大力支持,在此向他们表示衷心的感谢!

由于本系列手册的编译工作量大、时间紧,因此,尽管编译的老师和编译委员会都尽了最大的努力,但也难免有不妥,甚至错误之处,编译委员会全体老师恳切地希望广大读者给予批评指正。

清华大学生物医学工程系

胡广书 教授

2004 年 3 月

编 译 者 序

随着基于 DSP 的实时多任务应用的日益普遍,传统的围绕顺序处理和循环来构建 DSP 应用软件的开发途径已经不能够满足要求,越来越多的开发者开始采用基于 DSP 的多线程内核来设计应用软件。多线程能够使实时多任务应用程序被清晰地分解成多个功能不同、优先级不同的线程,所以一个多线程应用程序更容易在不影响某些关键线程的响应时间的前提下进行维护和升级,而这一点在采用顺序处理和循环结构的应用程序中则很难做到。

DSP/BIOS 是 TI 公司特别为 TMS320C6000™、TMS320C5000™和 TMS320C28x™系列 DSP 平台所设计开发的一个尺寸可裁剪的实时多任务操作系统内核,并且也能在 OMAP™器件的 DSP 核上运行。DSP/BIOS 已经过数以千计用户设计的验证,是 TI 公司的 Code Composer Studio™开发工具的组成部分之一,无须运行许可证,用户就可以免费使用。

通过使用 DSP/BIOS 提供的一系列丰富的内核服务,开发者能够快速创建满足实时性能要求的精细复杂的多任务应用程序。这些内核服务具有跨越 C6000、C5000 和 C28x DSP 平台的标准 API 接口,能被用户程序调用,易于移植。这些服务除了支持多线程调度管理外,还支持系统实时分析以及资源管理。DSP/BIOS 内核具有很大的尺寸伸缩性,多线程配置下的内核镜像的代码量最小仅有 1K 字(Word),占用的 DSP 资源非常少。

DSP/BIOS 和为 DSP 片上外设开发的芯片支持库(Chip Support Libraries)是兼容的,并被紧密地结合到 Code Composer Studio(CCS)交互式开发环境中。用户可以使用 CCS 中提供的配置工具裁剪 DSP/BIOS 内核镜像,建立多任务应用程序,然后使用 CCS 所提供的功能强大的内核对象观察器以及实时分析工具来调试多任务应用程序。

对于开发基于 TI DSP 的实时多任务应用软件的用户来说,了解和掌握 TI DSP/BIOS 内核原理和使用方法是非常重要的。目前 TI 网站上公布的 DSP/BIOS 内核的最新版本为 5.3,最新的有关 DSP/BIOS 实时内核的手册主要有:

TMS320 DSP/BIOS User's Guide, Literature Number: SPRU423F, Nov. 2004

——《DSP/BIOS 用户手册》描述了 DSP/BIOS 实时操作系统内核原理。

DSP/BIOS Driver Developer's Guide, Literature Number: SPRU616, Nov. 2002

——《DSP/BIOS 驱动开发手册》描述了 DSP/BIOS 设备驱动的开发和使用。

DSP/BIOS 5.3 Textual Configuration (Tconf) User's Guide, SPRU007H, May 2006

——《DSP/BIOS 文本配置用户手册》描述了用于配置 DSP/BIOS 应用程序的 JavaScript 脚本语言。

TMS320C6000 DSP/BIOS 5.31 API Reference, Literature Number: SPRU403N, Sep. 2006

TMS320C5000 DSP/BIOS 5.31 API Reference, Literature Number: SPRU404M, Sep. 2006

TMS320C28x DSP/BIOS 5.31 API Reference, Literature Number: SPRU625H, Sep. 2006

——《DSP/BIOS API 函数参考手册》详细描述了相应 DSP 平台: C6000、C5000、C28x 的 DSP/BIOS API 函数。

根据“TI DSP 系列中文手册编译委员会”提出的编译原则,我们对其中主要的“DSP/BIOS 用户手册(版本 E)”和“DSP/BIOS 驱动开发手册”进行了编译。在尊重原文的前提下,为了便于读者理解,我们对原文中某些部分内容叙述的顺序进行了调整、合并,对某些冗余的叙述进行适当删减。对于原文中个别叙述不清和错误之处进行了补充说明和订正。全书分为两个部分: DSP/BIOS 用户手册和 DSP/BIOS 驱动开发手册,共由 12 章和 4 个附录组成。第一部分共 7 章,具体内容包括 DSP/BIOS 概述、程序生成、监测、线程调度、存储器和低级函数、I/O 概述和管道、流 I/O 和设备驱动;第二部分共 5 章 4 个附录,具体内容包括设备驱动开发工具包(DDK)简介、DSP/BIOS 设备驱动的结构和使用、使用 DSP/BIOS 设备驱动、GIO 类驱动、微型驱动的开发步骤、IOM 接口、PIO 适配器、LIO 模型到 IOM 模型的移植、GIO API 的 ASYNC 扩展。

由于篇幅有限,对其他的 DSP/BIOS 技术手册没有进行编译。其中《DSP/BIOS 文本配置用户指南》为 DSP/BIOS 内核编程者提供了一种采用文本方式配置 DSP/BIOS 内核的方法。《DSP/BIOS API 函数参考手册》是本书的重要参考读物,在阅读本书时有可能需要查阅 API 函数参考手册以获得更多的信息。读者可以从 TI 的网站上下载这些手册进行查阅。另外,随着技术的发展,TI 的技术手册时有更新,请读者及时关注 TI 网站以获取 DSP/BIOS 内核的最新信息。

本书由西安电子科技大学通信工程学院王军宁教授主编和统稿,具体参加编译工作的有何迪、马娟、胡启龙等同志,其中王军宁编译第 1、8 章,何迪编译第 4、5、6、7 章,马娟编译第 9、10、11 章,胡启龙编译第 2、3、12 章,荣瓴、贾文编译组织附录和全书的校定工作,柯元旦参加了本书初稿的组织和部分章节的编译工作。另外特别感谢在本书编写过程中褚莉莉、罗建科、李鹏、杨林等同志的工作及清华大学出版社编辑们的大力支持和帮助。感谢 TI 公司大学计划和清华大学胡广书教授对于本书编译工作的指导和支持。

由于 DSP 技术发展迅速, TI DSP/BIOS 内核的版本也在不断升级,加上译者水平有限,书中可能存在对原著理解不妥及歧义、错误之处,我们恳切地希望广大读者给予批评指正。

译者

于西安电子科技大学通信工程学院

2007 年 2 月

目 录

第一部分 DSP/BIOS 用户手册

绪言	2
第 1 章 DSP/BIOS 概述	3
1.1 DSP/BIOS 的特色与优点	3
1.2 DSP/BIOS 组件	4
1.2.1 DSP/BIOS 实时内核和 API	5
1.2.2 DSP/BIOS 配置	6
1.2.3 DSP/BIOS 分析工具	7
1.3 命名规则	8
1.3.1 模块头文件名	9
1.3.2 对象名称	9
1.3.3 操作名	9
1.3.4 数据类型名	10
1.3.5 存储器段命名	11
1.3.6 标准存储段	12
1.4 更多的信息	13
第 2 章 程序生成	14
2.1 开发过程	14
2.2 静态配置 DSP/BIOS 应用程序	14
2.2.1 使用图形化配置工具	15
2.2.2 使用文本编辑器	15
2.2.3 配置 DSP/BIOS 应用程序的步骤简介	15
2.2.4 引用静态创建的 DSP/BIOS 对象	16
2.3 动态创建 DSP/BIOS 对象	19
2.4 建立 DSP/BIOS 程序使用的文件	21
2.5 编译和链接 DSP/BIOS 程序	22
2.5.1 构建 CCS 项目	22
2.5.2 使用 makefile 建立 DSP/BIOS 应用程序	23
2.6 DSP/BIOS 程序中的运行支持库	25

2.7	DSP/BIOS 启动序列	26
2.7.1	C5500 平台启动序列	28
2.8	DSP/BIOS 中使用 C++ 语言	28
2.8.1	存储器管理	29
2.8.2	名称改编	29
2.8.3	在配置中调用类的成员函数	30
2.8.4	类的构造函数和析构函数	30
2.9	DSP/BIOS 调用的用户函数	31
2.10	Main 函数中调用 DSP/BIOS API 函数	31
第 3 章	监测	33
3.1	实时分析	33
3.1.1	实时调试与循环调试的对比	33
3.1.2	软件监测与硬件监测的对比	33
3.2	监测性能	34
3.2.1	监测内核与非监测内核的对比	35
3.3	监测 APIs	36
3.3.1	显式监测与隐式监测的对比	36
3.3.2	事件日志管理器(LOG 模块)	37
3.3.3	统计对象管理器(STS 模块)	38
3.3.4	追踪管理器(TRC 模块)	42
3.4	隐式 DSP/BIOS 监测	44
3.4.1	执行图	44
3.4.2	CPU 负荷图	45
3.4.3	隐式 HWI 监测	47
3.4.4	最大堆栈深度	49
3.4.5	中断响应时间	50
3.5	内核对象观察	51
3.5.1	使用树型视图	52
3.5.2	使用右键快捷菜单	53
3.5.3	各种对象类型属性的显示	54
3.6	线程级调试	60
3.6.1	使能线程级调试	60
3.6.2	打开线程控制窗口	61
3.6.3	使用线程控制窗口	61
3.7	用于现场测试的监测	63

3.8 实时数据交换(RTDX)	63
3.8.1 RTDX 应用	63
3.8.2 RTDX 实例	64
3.8.3 RTDX 数据流	64
3.8.4 RTDX 运行模式	65
3.8.5 编写汇编代码时的特殊注意事项	66
3.8.6 RTDX 目标缓冲区大小	66
3.8.7 RTDX 数据的发送	66
第4章 线程调度	67
4.1 线程调度概述	67
4.1.1 线程类型	67
4.1.2 线程类型的选择	68
4.1.3 线程特点比较	69
4.1.4 线程优先级	70
4.1.5 让出和抢占	71
4.2 硬件中断	72
4.2.1 配置硬件中断	73
4.2.2 禁止和使能硬件中断	73
4.2.3 实时仿真模式对 DSP/BIOS 中断的影响	74
4.2.4 中断环境管理	77
4.2.5 寄存器	83
4.3 软件中断	84
4.3.1 创建 SWI 对象	84
4.3.2 在配置工具里设置软件中断优先级	85
4.3.3 软件中断优先级和应用程序堆栈大小	86
4.3.4 软件中断的执行	87
4.3.5 使用 SWI 对象的邮箱	87
4.3.6 使用 SWI 的优缺点	91
4.3.7 软件中断抢占时的寄存器保存	92
4.3.8 禁止和恢复 SWI	92
4.4 任务	93
4.4.1 创建任务对象	93
4.4.2 任务的执行状态和调度	95
4.4.3 检测堆栈溢出	97
4.4.4 任务钩子	97

4.4.5	用于额外环境保护的任务钩子	98
4.4.6	任务让出与时间片调度	99
4.5	空闲循环	101
4.6	功率管理	102
4.6.1	闲置时钟域	103
4.6.2	引导时节省功耗	104
4.6.3	电压和频率尺度调整	104
4.6.4	使用睡眠模式	105
4.6.5	睡眠及尺度调整的协调	105
4.7	信号灯	107
4.8	邮箱	112
4.9	定时器、中断和系统时钟	116
4.9.1	高分辨率和低分辨率时钟	116
4.9.2	系统时钟	118
4.9.3	系统时钟的实例	119
4.10	周期函数管理器(PRD)和系统时钟	120
4.10.1	调用 PRD 对象的函数	121
4.10.2	PRD 和 SWI 的统计信息	121
4.11	使用执行图观察程序的执行情况	122
4.11.1	执行图中的状态指示	122
4.11.2	执行图中的线程	123
4.11.3	执行图中的序列号	123
4.11.4	使用 RTA 控制面板设置执行图	124
第 5 章	存储器和低级函数	125
5.1	存储器管理	125
5.1.1	配置存储器段	126
5.1.2	禁止动态存储分配	126
5.1.3	在自己的链接命令文件中定义存储器段	127
5.1.4	动态存储分配	128
5.1.5	获得一个存储器段的状态	130
5.1.6	减小存储器碎片	130
5.1.7	MEM 模块使用举例	130
5.2	系统服务	134
5.2.1	停止程序执行	134
5.2.2	错误处理	135

5.3 队列	136
5.3.1 原子 QUE 函数	136
5.3.2 其他 QUE 函数	137
5.3.3 QUE 程序示例	138
第 6 章 I/O 概述和管道	141
6.1 I/O 概述	141
6.2 管道与流的对比	142
6.3 不同驱动模型的比较	143
6.3.1 创建一个使用 IOM 微型驱动的设备	143
6.3.2 创建一个使用 SIO 流和 DIO 适配器的设备	144
6.3.3 创建一个使用 SIO/DEV 模型的设备	144
6.3.4 创建一个使用 DSP/BIOS 提供的软件驱动的设备	145
6.4 数据管道管理器(PIP 模块)	145
6.4.1 写入数据到管道	146
6.4.2 从管道中读取数据	147
6.4.3 使用管道的通知函数	148
6.4.4 PIP 模块 API 函数的调用顺序	148
6.5 主机通道管理器(HST 模块)	150
6.5.1 传输 HST 数据到主机	151
6.6 I/O 性能问题	151
第 7 章 流 I/O 和设备驱动	153
7.1 流 I/O 和设备驱动概述	153
7.2 创建和删除流	155
7.2.1 静态创建流对象	155
7.2.2 动态创建和删除流对象	155
7.3 流 I/O——读入流和写出流	156
7.3.1 缓冲区交换	157
7.3.2 例子——从 DGN 设备读取输入缓冲区	158
7.3.3 例子——对 DGN 设备的读和写	160
7.3.4 例子——使用发放/回收模型的流 I/O	161
7.4 可堆叠设备	163
7.4.1 例子——SIO_create 和堆叠设备	164
7.5 流控制	168
7.6 流选择	169
7.6.1 程序示例	170

7.7	到多个客户端的流传输	170
7.8	主机与目标板之间数据的流传输	171
7.9	设备驱动模板	172
7.9.1	典型的文件组织	172
7.10	流 DEV 结构体	173
7.10.1	DEV_Fxns 结构体	173
7.10.2	DEV_Frame 结构体	173
7.10.3	DEV_Obj 结构体	174
7.11	设备驱动初始化	175
7.12	打开设备	176
7.13	实时 I/O	179
7.13.1	DEV_STANDARD 流传输模型	179
7.13.2	DEV_ISSUERECLAIM 流传输模型	180
7.14	关闭设备	181
7.15	设备控制	183
7.16	设备就绪	184
7.17	设备类型	186

第二部分 DSP/BIOS 驱动开发手册

绪言	190
第 8 章 设备驱动开发工具包简介	191
8.1 阅读指南	191
8.1.1 应用程序开发/整合者	191
8.1.2 驱动开发者	191
8.2 DSP/BIOS 驱动开发工具包概述	192
8.2.1 功能设备驱动	192
8.2.2 驱动模型	192
8.2.3 可复用的类驱动模块	193
8.3 DDK 工具包内容和组织	194
8.4 使用应用程序示例	195
第 9 章 DSP/BIOS 设备驱动的结构和使用	197
9.1 设备驱动双层模型	197
9.1.1 应用程序结构概述	198
9.1.2 驱动的初始化和绑定	198
9.1.3 设备实例和通道实例	199

9.2 设备驱动数据流	199
9.2.1 通道实例句柄	200
9.2.2 IOM 请求包	200
9.2.3 通道操作	202
9.2.4 I/O 请求的递交	202
9.2.5 设备控制	203
9.3 类驱动概述	203
9.3.1 SIO 适配器(DIO)	203
9.3.2 PIP 适配器(PIO)	204
9.3.3 GIO 类驱动	205
第 10 章 使用 DSP/BIOS 设备驱动	206
10.1 注册微型驱动	206
10.2 配置 SIO/DIO 类驱动	207
10.2.1 应用程序示例	207
10.2.2 配置过程	208
10.3 配置 PIP/PIO 类驱动	209
10.3.1 应用程序示例	209
10.3.2 配置过程	209
10.4 配置 GIO 类驱动	211
10.4.1 应用程序示例	211
10.4.2 配置过程	211
第 11 章 GIO 类驱动	212
11.1 GIO 模块概述	212
11.2 GIO 实现细节	213
11.2.1 GIO_Obj 结构体	213
11.3 错误处理	214
11.4 扩展 GIO 的 API	214
第 12 章 微型驱动的开发步骤	216
12.1 微型驱动的设计与实现	216
12.1.1 绑定通道——mdBindDev	216
12.1.2 创建和删除通道——mdCreateChan/mdDeleteChan	217
12.1.3 递交 I/O 请求——mdSubmitChan	218
12.1.4 服务设备中断并完成 I/O 操作——ISR	218
12.1.5 控制设备——mdControlChan	218

12.2	C5402 SBS 微型驱动示例	219
12.2.1	常量、类型和结构	219
12.2.2	初始化函数	220
12.2.3	mdBindDev 函数	221
12.2.4	mdControlChan 函数	223
12.2.5	mdCreateChan 函数	224
12.2.6	mdDeleteChan 函数	225
12.2.7	mdSubmitChan 函数	226
12.2.8	mdUnBindDev 函数	227
12.2.9	ISR 函数	228
附录 A	IOM 接口	230
A.1	微型驱动接口概述	230
附录 B	PIO 适配器	237
B.1	PIO 适配器接口概述	237
附录 C	LIO 模型到 IOM 模型的移植	244
C.1	LIO 模型和 IOM 模型的比较	244
C.1.1	LIO 概念回顾	244
C.1.2	LIO 适配器与 IOM 类驱动的对比	245
C.1.3	LIO 接口函数与 IOM 接口函数的对比	245
C.2	LIO 模型到 IOM 模型的移植	246
C.2.1	配置	246
C.2.2	初始化	246
C.3	LIO 控制器到 IOM 微型驱动的移植	246
附录 D	GIO API 的 ASYNC 扩展	248
D.1	ASYNC 模块概述	248
参考文献		256