

HZ Books



中国电子学会嵌入式专家委员会指定教材

Windows CE.Net 程序设计

姜波 编著



机械工业出版社
China Machine Press



中国电子学会嵌入式专家委员会指定教材

TP316.7

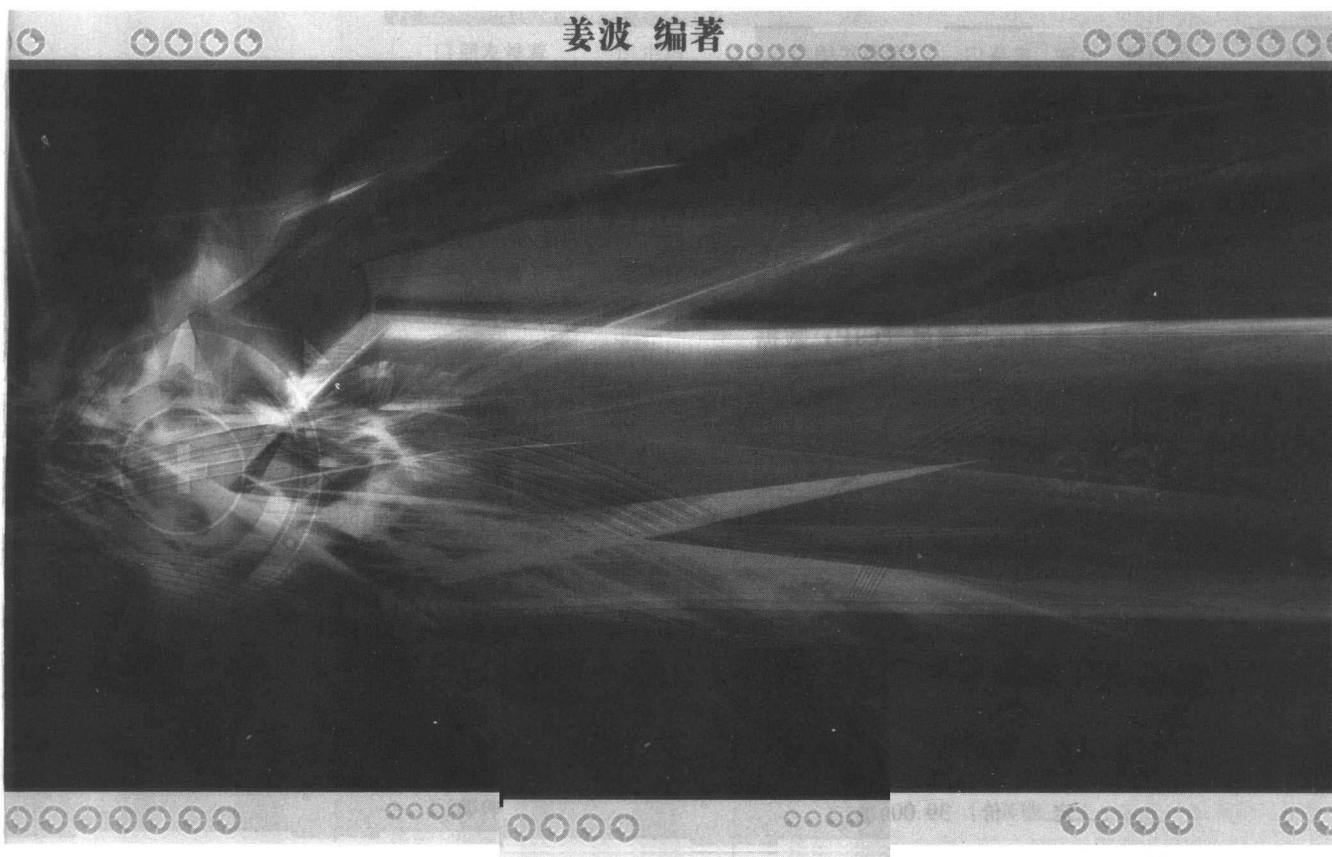
113

2007

Windows CE.Net

程序设计

姜波 编著



机械工业出版社
China Machine Press

本书系统而全面地介绍了 Windows CE .Net的基础理论和特点,以及相关的程序设计方法。本书给出了一定数量的程序源代码及代码讲解,力求在讲授基础理论的同时,最大限度地提高读者的实际动手能力。本书主要分为三个部分。第一部分主要介绍 Windows CE .Net程序设计的基础知识,包括 Windows CE .Net的启动、开发工具、内存管理、文件管理、系统输入等。第二部分主要介绍高级 Windows CE .Net程序设计,包括中断处理、驱动程序模型、流驱动程序开发等。第三部分为开发实例。

本书语言通俗易懂,叙述讲解由浅入深,即可作为 Windows CE .Net程序设计的参考用书,也可作为各高校及培训机构的授课用书。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

Windows CE .Net程序设计/姜波编著. —北京:机械工业出版社, 2007. 1

ISBN 7-111-20121-3

I. W… II. 姜… III. 窗口软件, Windows CE—程序设计 IV. TP316.7

中国版本图书馆 CIP 数据核字 (2006) 第 123816 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:杨庆燕

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2007 年 1 月第 1 版第 1 次印刷

186mm×240mm·24.75 印张

定 价: 39.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线:(010) 68326294

专家指导委员会

(按姓氏笔画顺序)

主任：王越（院士）

副主任：王田苗（教授） 魏洪兴

委员：马洪连 王磊 田景文 关永
张国印 闵华松 严义 邵子立
邱秀芹 周亦敏 杨刚 贺平
贾智平 康一梅 章登义 潘巨龙
魏洪兴 魏淑桃

前 言

20 世纪，以 PC 机为代表的信息技术的迅速发展，对世界的政治、经济、军事和科教等方面产生了深刻的影响；由此而产生的信息产业已经成为世界经济支柱之一。如果说 20 世纪被称之为“PC 时代”的话，那么 21 世纪则被称之为“后 PC 时代”。而嵌入式技术无疑是这个时代最具代表性的技术之一。

众所周知，在信息技术领域，我们和西方发达国家相比还存在一定的差距。嵌入式技术和产品的大规模应用，为我们提供了一个缩小差距的机会。在这个以嵌入式技术为代表性技术之一的“后 PC 时代”，培养出具备专业技能，同时可以适应国际竞争需要的嵌入式专业人才就显得尤为重要。在这种理念的指引下，中国电子学会及中国电子学会嵌入式专家委员会，在吸收国内外先进教学经验的基础上，结合我国实际情况，推出了此套嵌入式认证考试系列教材。其宗旨是普及嵌入式技术的基础知识，提高从业人员的实际动手能力，为嵌入式技术的发展和普及，为提高我国嵌入式技术的水平，进行有益的探索和尝试。

实际上，从消费类电子产品领域，到航空航天，汽车电子，再到我们日常使用的网络通信、数据传输等领域，都有着嵌入式技术的身影。随着嵌入式技术的不断发展，以及其应用领域的日益广泛，嵌入式产品也在不断地改变着我们的生活方式。

首先从体系结构上看，嵌入式产品一般都是由硬件和软件所组成。比如我们使用的手机、MP3 等产品。其次从开发角度上看，嵌入式产品的设计往往需要专用的开发工具；比如本书所涉及的 Windows CE .Net 程序设计，我们用到了 Embedded Visual C++ 和 Platform Builder 等开发工具。再次从应用领域上看，那些使用 PC 机的地方几乎都可以用相应的嵌入式产品所替代。同时，嵌入式产品的广泛运用还极大地提高了整个系统的实时性。我们可以在不同的位置安放大量小型嵌入式产品，来实时地获取周围环境的参数，取代过去那种分散数据采集，再由中央计算机集中处理的方式。最后从用户体验上看，嵌入式产品性能的日益提高，给我们的工作、生活带来便利，提高了产品的使用感受。以目前日益流行的智能手机为例，其 CPU 的频率早已经达到了 400MHz，最先进的产品已经达到了 600MHz。同时还出现了运行在手机上的移动显卡等，这些都使得我们的手机在具备传统语音通话功能的同时，还具备了强大的多媒体播放功能、丰富的游戏功能、实时的网络浏览功能等。或许有一天，我们出差在外的时候，随身只要挟带一部移动电话，就可以实现文字办公、网络浏览、邮件发送、视频播放等诸多功能。

在嵌入式产品中，广泛应用的嵌入式操作系统主要有嵌入式 Linux、Windows CE、VxWorks、Symbian 等。而本书介绍的 Windows CE .Net 是 Windows CE 应用最为广泛的版本之一，Windows CE 在高端智能手机领域占据了极其重要的市场地位及市场份额。Windows CE .Net 是一个具有抢先式多任务功能，并具有强大通信能力的嵌入式操作系统。也是微软公

司专门为移动设备和消费类电子产品、嵌入式应用等非 PC 领域而全新设计的操作系统产品。因此根据其应用环境的特点, Windows CE .Net被设计成具有高度模块化、良好实时性、强大通信能力、支持多种 CPU 的嵌入式操作系统。它主要具有以下特点:

1) 良好的电源管理功能。由于很多嵌入式产品都是依靠电池来供电的, 因此作为一款嵌入式操作系统, Windows CE .Net具备灵活的电源管理功能, 包括睡眠/唤醒模式等。

2) 良好的多任务管理功能。与桌面 Windows 类似, Windows CE .Net同样具备良好的多任务管理功能。我们可以同时运行多个程序, 并在这些程序之间自由地切换。

3) 人性化的图形用户界面。Windows 获得成功的原因之一在于其使用了友好的图形用户界面。Windows CE .Net同样具备了类似的功能, 当我们启动它的时候, 不但会看到我们熟悉的操作系统界面, 其使用方式和文件命名方式同桌面 Windows 也是完全一样的。

4) 多样的通信功能。Windows CE .Net的通信模块主要支持串行端口通信、红外通信和网络通信。我们可以利用这些功能灵活地设计出符合实际需要的产品来。

5) 优秀的多媒体扩展性。可以通过内置的 MediaPlayer 播放器播放音频和视频文件, 同时还加入了对 SD 卡、CF 卡等移动存储设备的支持。

6) 强大的实时功能。由于 Windows CE .Net是一款实时性操作系统, 因此可以被应用在很多对实时性要求较高的领域。

那么如何设计 Windows CE .Net下的应用程序, Windows CE .Net具有哪些特点, 它和传统的桌面 Windows 相比又有怎样的不同呢? 这就是本书要解决的问题。

在本书中, 作者全面、系统地介绍了 Windows CE .Net程序设计的基本知识及程序设计方法。不仅在理论上详细地介绍了 Windows CE .Net的特点, 还通过多个程序实例的构建和代码讲解, 向读者介绍了使用 Embedded Visual C++ 和 Visual Studio .Net 2003 进行 Windows CE .Net程序设计的基本步骤和方法。使读者在了解基础理论的同时, 进一步提高了自己的实际动手能力。同时, 本书一改技术类书籍语言晦涩难懂的现状, 做到了叙述语言的通俗易懂。

本书共分为三大部分 18 章, 具体内容安排如下:

第一部分介绍 Windows CE .Net程序设计的基础知识, 包括第 1 章到第 9 章。包括什么是 Windows CE .Net, Windows CE .Net的特点和用途是什么, 启动过程和开发工具介绍, Windows CE .Net下的内存管理及进程、线程的特点, 以及文件系统和注册表管理。同时在这里我们还为读者介绍了如何使用相关的 API 函数进行 Windows CE .Net下屏幕图形的绘制和实现系统的输入。通过这一部分的学习, 读者将会对 Windows CE .Net有一个感性的认识。

第二部分介绍 Windows CE .Net驱动程序设计, 包括第 10 章到第 15 章。主要介绍 Windows CE .Net中的中断处理, 驱动程序模型, 流驱动程序开发以及不同总线的特点。同时在这一部分中, 我们还为读者介绍了 Windows CE .Net下 PS/2 键盘驱动程序设计和触摸屏等相关内容。

第三部分为开发实例, 包括第 16 章到第 18 章。主要介绍如何使用 Platform Builder 进行 Windows CE .Net的内核定制, Platform Builder 有哪些特点, 如何使用 Embedded Visual C++ 开发 Windows CE .Net程序以及如何使用 Visual Studio .Net 2003 开发 Pocket PC 版俄罗斯方

块。在这里不但有详细的操作流程，我们还为读者详细地介绍了程序代码。使得读者不但可以按照我们叙述的步骤按部就班地操作，还可以对程序代码有非常深刻的认识。

我在进行 Windows CE .Net相关软件的开发过程中，深切地体会到，只有不断地实践才会把所学到的知识很好地运用到实际工作中。一些在书本上看似简单的原理，在应用中我们才会体会到其精髓所在。因此，在这里建议读者在阅读本书的同时，最好可以按照书中的操作步骤，亲自把整个程序构建一次；同时还可以尝试修改程序中的某段代码，看看其效果和原来有何不同，或者把自己的想法和疑问发到我们的网站上，和大家进行交流。我想，这些都会加深读者对 Windows CE .Net程序设计的理解，真正提高大家的动手能力和思考问题的能力。

本书在撰写的过程中，得到了众多良师益友的大力支持。我的硕士研究生导师，北京航空航天大学的魏洪兴副教授，不但向我传授嵌入式程序设计的相关知识，还积极地协调各方面的资源，保证了本书的顺利完成；北京博创兴业科技有限公司的相关技术人员也为本书提供了大量的技术支持。

本书主要由姜波编写，参加编写的还有李淑云、宁峰、马一宁、杨亮和邹莹。

在这里还要特别感谢我的父母，是他们在本书的编写过程中给了我极大的鼓励与支持，感谢他们为我所做的一切。

由于作者水平有限，加之时间仓促，书中难免出现不足之处，请各位读者批评指正。如果有什么疑问，可以访问 www.deegua.com 或者发送电子邮件到 deegua@126.com。

姜 波

2006年10月于北京

目 录

专家指导委员会

前 言

第一部分 初级 Windows CE .Net 程序设计

第 1 章 嵌入式系统简介	1
1.1 什么是嵌入式系统	1
1.1.1 嵌入式系统的特点	2
1.1.2 嵌入式系统软件的特点	2
1.1.3 嵌入式系统的分类	3
1.2 嵌入式系统的组成	4
1.2.1 嵌入式系统处理器的分类	4
1.2.2 嵌入式微处理器的主要系列	5
1.2.3 嵌入式系统的存储器	9
1.2.4 嵌入式系统的输入方式	10
1.3 嵌入式系统的应用领域	12
1.3.1 在消费类电子产品中的应用	12
1.3.2 在工业控制领域的应用	13
1.3.3 在通信领域的应用	13
1.3.4 在机器人、办公等其他领域的 应用	14
1.4 几种常见的嵌入式操作系统	15
1.4.1 嵌入式操作系统的分类	15
1.4.2 Windows CE	16
1.4.3 Symbian	17
1.4.4 Palm OS	18
1.4.5 嵌入式 Linux	21
1.4.6 VxWorks	22
第 2 章 Windows CE .Net简介	26
2.1 Windows CE 的发展历史	26
2.2 Windows CE 功能简介	27
2.2.1 Windows CE 的电源管理功能	28

2.2.2 Windows CE 的多任务功能	28
2.2.3 Windows CE 的图形界面	28
2.2.4 Windows CE 的通信能力	29
2.2.5 Windows CE 的多媒体和 扩展功能	29
2.2.6 Windows CE 的实时功能	30
2.3 Windows CE .Net的体系结构	30
2.3.1 Windows CE .Net的层次架构	30
2.3.2 内核模块	31
2.3.3 文件系统	32
2.3.4 网络与通信模块	32
2.3.5 GWES 模块	33
2.4 Windows CE 的应用领域	34
2.4.1 Windows CE 在移动计算领域的 应用	34
2.4.2 Windows CE 在嵌入式领域的 应用	37
第 3 章 Windows CE .Net开发基础	38
3.1 Windows CE .Net的启动过程	38
3.1.1 系统复位	38
3.1.2 运行 KernelStart	38
3.1.3 运行 FileSys . exe	39
3.1.4 运行可选择的进程	39
3.1.5 运行 shell . exe和 device . exe	41
3.1.6 运行 gwes . exe	42
3.1.7 运行自定义进程	43
3.1.8 运行 Explorer . exe进程	43
3.2 Windows CE 设备的电源特性	44
3.3 Windows CE .Net程序开发环境	44
3.3.1 Platform Builder	45
3.3.2 EVC 和 EVB	46
3.3.3 Visual Studio .Net	47
3.3.4 ActiveSync	48

3.3.5 模拟器	48	5.3 堆	85
3.3.6 程序开发所需的硬件	50	5.3.1 Windows CE 中堆的特性	85
3.4 Windows CE .Net目录解析	52	5.3.2 本地堆	86
3.4.1 Windows CE 安装过程中		5.3.3 独立堆	87
应该注意的问题	52	5.4 栈	89
3.4.2 目录解析	52	5.5 静态数据块	89
第 4 章 Windows CE .Net开发工具		5.6 Windows CE 下的 Bootloader	90
简介	54	5.6.1 Bootloader 简介	90
4.1 Platform Builder	54	5.6.2 Bootloader 的主要功能	91
4.1.1 Platform Builder 简介	54	5.6.3 Bootloader 的结构	91
4.1.2 使用 Platform Builder 进行		第 6 章 Windows CE .Net中的进程	
平台的定制	57	和线程	93
4.2 Embedded Visual C++	62	6.1 Windows CE .Net中的进程	93
4.2.1 Embedded Visual C++简介	62	6.1.1 进程概述	93
4.2.2 Win32 API、MFC 及 ATL		6.1.2 创建进程	94
三种程序设计方式	65	6.1.3 终止进程	96
4.2.3 在 Embedded Visual C++中		6.1.4 其他相关函数	97
构建一个 MFC 程序	66	6.2 Windows CE .Net中的线程	98
4.2.4 在构建好的 MFC 程序		6.2.1 线程概述	98
框架中添加消息	70	6.2.2 创建线程	100
4.2.5 EVC 在安装及使用过程中		6.2.3 设置线程的优先级	101
应该注意的问题	72	6.2.4 查询线程的优先级	101
4.3 Visual Studio .Net	72	6.2.5 设置线程的时间片	102
4.3.1 Visual Studio .Net 2003 简介	73	6.2.6 挂起和恢复一个线程	102
4.3.2 使用 Visual Studio .Net 2003		6.3 进程间通信	102
构建 Smart Device 工程	74	6.3.1 事件对象	103
第 5 章 Windows CE .Net中的内存		6.3.2 线程等待	104
管理	77	6.3.3 信号量	105
5.1 RAM 和 ROM	77	6.3.4 互斥	106
5.1.1 RAM	77	6.3.5 互锁函数	107
5.1.2 ROM	78	6.3.6 临界区	108
5.1.3 RAM 和 ROM 在实际设备		第 7 章 Windows CE .Net中的文件	
中的体现	78	系统和注册表管理	110
5.2 虚存	80	7.1 Windows CE .Net文件系统	110
5.2.1 Windows CE .Net中的虚拟		7.1.1 文件系统概述	110
内存结构	80	7.1.2 对象存储	111
5.2.2 Windows CE .Net中的分页式		7.1.3 文件的输入输出	112
虚拟存储	81	7.1.4 内存映射文件	117
		7.1.5 文件管理	120

7.2 注册表	123	9.2.1 输入焦点	156
7.2.1 概述	123	9.2.2 键盘消息	157
7.2.2 相关 API 函数	124	9.2.3 键盘测试	160
第 8 章 Windows CE .Net 中的屏幕		9.3 鼠标输入	160
绘制函数	128	9.4 Windows CE 中的触摸屏	161
8.1 Hello Windows CE	128	9.4.1 手写笔消息	161
8.1.1 程序构建过程	128	9.4.2 电子墨水	162
8.1.2 代码添加	133	9.4.3 关于输入焦点	162
8.2 文本显示相关的函数	133	9.4.4 右键单击	163
8.2.1 DrawText 函数	134	9.5 通过鼠标点击的方式进行屏幕绘制	164
8.2.2 ExtTextOut 函数	134	9.5.1 程序实例中涉及的主要函数	164
8.2.3 文本的属性	135	9.5.2 程序实例的运行效果	165
8.2.4 SetBkColor 函数	136	9.5.3 程序实例的完整代码	166
8.2.5 SetBkMode 函数	136	9.6 在 Windows CE 中实现鼠标	
8.3 字体	137	移动轨迹的绘制	170
8.4 图形绘制	138	9.6.1 程序实例中涉及的主要函数	170
8.4.1 直线的绘制	138	9.6.2 程序实例的运行效果	172
8.4.2 画笔	139	9.6.3 程序实例的完整代码	172
8.4.3 刷子	140	第二部分 高级 Windows CE	
8.4.4 矩形	140	.Net 程序设计	
8.4.5 圆	141	第 10 章 Windows CE .Net 中的	
8.4.6 圆角矩形	142	中断处理	177
8.5 使用 Win32 API 编程方式在		10.1 中断基础知识	177
Windows CE 中显示文本	142	10.1.1 汇编语言与微处理器的	
8.5.1 程序实例中涉及的框架函数	143	体系结构	177
8.5.2 程序实例中涉及的文本		10.1.2 中断处理程序	180
显示函数	144	10.1.3 保存和恢复上下文	181
8.5.3 程序实例的运行效果	145	10.1.4 中断禁止和中断延迟	182
8.5.4 程序实例的完整代码	146	10.1.5 共享数据问题的产生	183
8.6 使用 Win32 API 编程方式在		10.1.6 共享数据问题的解决	186
Windows CE 中绘制图形	149	10.2 Windows CE .Net 中断的组成	186
8.6.1 程序实例中涉及的主要函数	149	10.2.1 中断服务例程 ISR	187
8.6.2 程序实例的运行效果	151	10.2.2 中断服务线程 IST	188
8.6.3 程序实例的完整代码	152	10.3 Windows CE .Net 中断处理	
第 9 章 Windows CE .Net 系统		机制的特点	189
中的输入	155	10.4 Windows CE .Net 中断发生和	
9.1 概述	155	处理的过程	189
9.2 键盘输入	155	10.5 Netarm2410-S 设备中断简介	191

第 11 章 Windows CE .Net 中的驱动

程序	194
11.1 Windows CE .Net 的驱动程序模型	194
11.1.1 什么是驱动程序	194
11.1.2 本机驱动程序	195
11.1.3 流接口驱动程序	195
11.1.4 Windows CE .Net 中的驱动 程序接口	196
11.2 驱动程序的组成	196
11.2.1 单体驱动程序	197
11.2.2 分层驱动程序	197
11.2.3 单体/分层驱动程序在 Windows CE .Net 中的集成	197
11.3 Windows CE .Net 中的类型 驱动程序	199
11.4 Windows CE 简单驱动开发实例	199
11.4.1 原理及说明	199
11.4.2 步骤 1: 建立模拟器工具	201
11.4.3 步骤 2: 创建驱动程序工程	201
11.4.4 步骤 3: 编写驱动程序代码	203
11.4.5 步骤 4: 准备配置文件	205
11.4.6 步骤 5: 编译内核	206
11.4.7 步骤 6: 加载驱动	208

第 12 章 Windows CE .Net 中的流驱动

开发和相关注册表项	209
12.1 流驱动程序简介	209
12.1.1 什么是流接口驱动程序	209
12.1.2 流接口驱动程序的系统结构	210
12.2 设备文件名	211
12.2.1 设备文件名的格式	211
12.2.2 设备文件名的前缀	211
12.2.3 设备文件名的索引	212
12.3 流驱动的标准接口	212
12.3.1 常用的接口函数	212
12.3.2 接口函数的函数原型	213
12.3.3 与应用程序之间的调用关系	215
12.3.4 文件操作和设备操作函数 参数间的对应关系	216
12.3.5 单访问和多访问	218

12.4 样本流接口驱动程序	219
12.5 Windows CE .Net 的注册表	219
12.5.1 Windows CE .Net 注册表的 结构	220
12.5.2 Windows CE .Net 的常用键 值类型	221
12.5.3 流驱动程序用到的注册表 设置	221
12.5.4 在驱动程序中访问注册表	223
12.6 流接口驱动程序的加载和卸载	225
12.6.1 流接口驱动程序的加载	225
12.6.2 流接口驱动程序的卸载	226
12.7 ADC 流驱动程序	226
12.7.1 A/D 转换器简介	226
12.7.2 A/D 转换器的重要指标	228
12.7.3 Arm 自带的十位 A/D 转换器	229
12.7.4 ADC 驱动程序设计	231
12.7.5 操作步骤	233

第 13 章 Windows CE .Net 中的总线

13.1 总线概述	235
13.2 总线的分类和主要参数	236
13.2.1 总线的分类	236
13.2.2 总线的主要参数	237
13.3 内部集成 (IIC) 总线	237
13.3.1 内部集成 (IIC) 总线的特点	238
13.3.2 传输方式	238
13.3.3 内部集成总线 (IIC) 与硬件 设备之间的数据交换	239
13.4 PCI 总线	242
13.4.1 PCI 总线的主体性能及特点	242
13.4.2 PCI 总线的外设自动识别	243
13.4.3 Windows CE .Net 中 PCI 总线 驱动的作用	243
13.4.4 PCIBus.dll 对 PCI 驱动的 加载	244
13.5 USB 总线	244
13.5.1 USB 结构的特点	244
13.5.2 USB 设备的结构	245
13.5.3 USB 设备的传输类型	246
13.5.4 USB 设备的软件设计	247

13.5.5	Windows CE .Net下 USB 系统的组成	247	16.2.2	选择板级支持包	270
13.5.6	Windows CE .Net下 USB 设备驱动程序的编写	247	16.2.3	平台配置	272
第 14 章 Windows CE .Net下的 PS/2 键盘驱动程序设计			16.2.4	自定义设备	273
14.1	PS/2 电气接口概述	250	16.2.5	选择程序开发的类库	275
14.2	键盘及键盘协议	251	16.2.6	选择应用程序	276
14.2.1	键盘的扫描码集	251	16.2.7	选择操作系统的核心服务	278
14.2.2	虚拟键码	252	16.2.8	连接服务和网络服务	280
14.3	键盘驱动程序的工作流程	252	16.2.9	选择文件系统和数据存储方式	281
14.3.1	键盘驱动程序的主要接口	253	16.2.10	字体的选择	282
14.3.2	PDD 层的开发	254	16.2.11	选择所支持的国家	282
14.3.3	一个典型的 IST 工作过程	256	16.2.12	选择客户端的 Internet 服务	284
14.3.4	键盘驱动程序结构图	256	16.2.13	多媒体技术的选择	284
第 15 章 Windows CE .Net下的 触摸屏			16.2.14	安全设置	286
15.1	触摸屏概述	258	16.2.15	选择用户界面	287
15.2	触摸屏的工作原理	258	16.2.16	提示信息	287
15.3	电阻类触摸屏	259	16.2.17	完成操作系统镜像的建立	288
15.3.1	触摸屏的导电涂层	259	16.3	使用 Platform Builder 生成操作系统镜像	288
15.3.2	五线电阻触摸屏	260	16.4	把操作系统镜像下载到模拟器	292
15.4	触摸屏与 Windows CE .Net系统	260	第 17 章 用 Embedded Visual C++进行 Windows CE 程序设计		
15.4.1	触摸屏设置函数	261	17.1	Embedded Visual C++介绍	295
15.4.2	触摸屏使能函数	261	17.1.1	EVC 4.0 特性简介	295
15.4.3	坐标转换函数	262	17.1.2	EVC 中的模拟器	298
15.4.4	采样坐标点函数	262	17.1.3	EVC 中的远程调试工具	300
第三部分 开发实例			17.2	用 EVC 编写俄罗斯方块步骤	304
第 16 章 用 Platform Builder 进行 内核定制			17.2.1	MFC 程序设计和 Win32 程序设计不同	304
16.1	Platform Builder 简介	265	17.2.2	程序简介	305
16.1.1	Platform Builder 的主要功能	265	17.2.3	工程的建立	306
16.1.2	Platform Builder 的用户界面	266	17.2.4	代码的添加	310
16.1.3	定制嵌入式系统平台主要包含的内容	269	17.3	游戏程序界面的设计	329
16.2	使用 Platform Builder 建立操作系统镜像	270	17.4	源代码解释	332
16.2.1	进入新建 Platform 向导	270	17.4.1	fanguaiDlg.h 代码注释	332
			17.4.2	fanguaiDlg.cpp 代码注释	333
			第 18 章 用 Visual Studio 进行 Pocket PC 程序设计		
			18.1	Pocket PC 简介	348

18.1.1	PDA 的发展历史	348	18.6.1	全局量的声明	373
18.1.2	Palm OS 操作系统	352	18.6.2	相关按键和菜单的处理	374
18.1.3	Windows CE 操作系统	353	18.6.3	方块的下落	375
18.1.4	Symbian	354	18.6.4	游戏的初始化	375
18.1.5	Linux 操作系统	354	18.6.5	游戏的重置	375
18.2	Pocket PC 的发展	355	18.6.6	游戏开始	376
18.2.1	硬件的发展趋势	355	18.6.7	游戏的暂停和退出	376
18.2.2	图形控制器	356	18.6.8	进行游戏数据的加载	376
18.2.3	I/O 设备	356	18.6.9	方块的加载	377
18.2.4	智能设备的供电解决方案	358	18.6.10	方块的放置	377
18.2.5	网络技术的应用	359	18.6.11	方块的旋转	378
18.3	Visual Studio. Net 2003 简介	360	18.6.12	方块的移动和落定函数	379
18.4	程序的构建过程	362	18.6.13	判断移除整行	379
18.5	源代码的添加	364	18.6.14	分数和等级的设定	381
18.6	源代码的注释	373	18.6.15	程序画面的实现	381



第一部分

初级 Windows CE .Net程序设计

第 1 章 嵌入式系统简介

1.1 什么是嵌入式系统

在 IT 行业，如果把公元 2000 年作为科技史的一个分水岭的话，那么 2000 年之前可以被称之为“PC”时代；而 2000 年之后则要被称为“后 PC”时代。这是因为在后 PC 时代，多种多样的嵌入式产品已经开始逐渐取代 PC 机，成为人们处理信息的工具。小到风头正劲的 MP3、手机、数码相机等常用的消费类电子产品，大到网络家电、智能家电、车载电子设备（如 GPRS 导航仪等）、工业机器人以及智能玩家……都在不知不觉中形成了一个充满商机的巨大产业，并且正在成为整个软件行业的支柱。那么到底什么是嵌入式系统呢？它和我们经常使用的 PC 系统有什么相同之处，又有什么不同之处呢？下面的内容将为您揭开这些问题的答案。

实际上，在当今的世界中，很多东西都是依靠嵌入式计算机而运行的。现代生活的各个领域，从我们的日常生活用品到交通工具等都有嵌入式计算机的身影。比如我们使用的手机就是一个典型的嵌入式计算设备，它里面包含了处理器、各种应用软件以实现通信等相关功能。另外数码相机、日常乘坐的汽车以及各种工业控制装置中都“嵌入”了各种嵌入式系统。根据 IEEE 的定义，嵌入式系统是“控制、监视或者辅助设备、机器、运行和车间运行的装置”。通俗地讲，嵌入式系统就是以应用为中心、以计算机技术为基础的专用计算机系统。它是与我们经常提到的桌面 PC、小型机和大型机相对而言的。通常情况下，嵌入式系统不以独立设备的形式出现，其大部分或者全部系统都嵌入在其他设备中，根据主体设备的需要来发挥自己的作用。

尽管嵌入式系统软件设计正日益风行，但是从 20 世纪 70 年代单片机的出现到今天各种嵌入式设备的大规模应用，嵌入式系统已经有了 30 多年的历史。和我们常用的桌面 PC 系统一样，嵌入式系统的发展也是伴随着嵌入式软件和硬件的发展而迅速壮大起来的。

说到嵌入式系统，最早是基于单片机的。20 世纪 70 年代单片机的大规模应用使得家电制



造、工业设备以及其他各种商品得以通过内置的电子装置获得迅速的普及。当时那些内置了电子装置的设备已经具备了嵌入式系统的一些特点。

进入到 20 世纪 80 年代，一些嵌入式系统的程序设计人员开始进入到嵌入式软件的编写领域，这就大大地提高了嵌入式软件的开发效率，同时也促进了嵌入式硬件设备的发展。

跨入到 20 世纪 90 年代以后，随着各行业对实时设备的需求越来越大，实时性的嵌入式操作系统被应用到越来越多的领域。此时，越来越多的公司看到了嵌入式系统良好的市场前景，纷纷加入到这个领域中来。市场上出现了 Palm OS、Windows CE、嵌入式 Linux 等多种嵌入式操作系统。嵌入式系统正式进入到高速发展时期。

1.1.1 嵌入式系统的特点

前面我们已经介绍了嵌入式系统的定义，下面就让我们来看一下嵌入式系统具有哪些特点。

- 嵌入式系统在通常情况下是面向特定的用户和特定的行业的。因此其设计和开发必须和行业特点结合在一起。
- 嵌入式系统的 CPU 等硬件设备必须具有体积小、功耗低、集成度高的特性，只有这样才能把很多功能集成在一个嵌入式芯片的内部。
- 嵌入式系统是计算机技术、半导体技术以及电子技术……与各个行业特点和行业应用结合在一起的产物。
- 嵌入式系统的软件和硬件都具有高效、简洁的特性。同时由于嵌入式设备的硬件性能有限，因此需要对嵌入式设备的软件和硬件进行一定的裁剪，以便做到量体裁衣。
- 嵌入式系统具有较高的可靠性、较低的成本和较低的耗电量。在一些特殊的领域中，实时性是对嵌入式系统的基本要求。
- 嵌入式系统由于其自身的特殊性，因此需要特殊的开发工具和程序设计环境。在本书中我们就使用的是 Embedded Visual C++ 和 Platform Builder。
- 嵌入式系统在通常情况下是和具体的产品结合在一起的。因此嵌入式系统的升级换代往往是伴随着所附着产品的升级换代而完成的。即使是同一种产品，只要产品的硬件指标不同（例如有的芯片是 ARM，有的是 Intel 的 XScale），一般都需要对软件进行相应的移植。

1.1.2 嵌入式系统软件的特点

和传统的软件设计相比，嵌入式系统软件开发具备了一些独有的特点。

- 关于系统的响应时间：由于嵌入式系统大多应用在对实时性要求较高的场合，因此系统要能够在极短的时间内做出对事件和信号的响应。
- 关于软件的成本：在一般情况下，嵌入式系统作为整个产品的一部分，其自身成本直接关系到整个产品的成本。因此通常情况下，嵌入式软件的功能做到刚刚“够用”即可，而不像 Windows 应用软件功能越多越好。例如我们经常使用的自动取款机，显然是不需要提供多媒体播放功能的。
- 关于系统的能耗：有很多嵌入式设备都是依靠电池供电的，因此嵌入式软件要尽可能小

地消耗系统的能源，以达到节省电能的目的。

- 关于软件的可靠性：因为很多的嵌入式设备都是在无人看管的情况下运行的（很少需要我们时刻注意软件的性能，比如手机，只有我们在打电话的时候才会注意到手机的信号等性能指标。而在待机状态我们则很少查看手机信号的强弱），因此软件运行时的可靠性和稳定性就尤为重要。尤其是当嵌入式软件被应用在一些大型设备的关键部位的时候，可靠性的好坏往往会关系到整个系统的安全性、稳定性等诸多指标。
- 关于软件占用的内存空间：由于嵌入式系统的特殊性，有很多嵌入式设备的内存都非常小，例如常用的 Pocket PC 其内存空间大多是 64MB 和 128MB，这样的空间大小和我们的桌面 PC 系统动辄几百兆字节，甚至上 G 的内存相比，空间可以称得上是少得可怜了。正是因为这样的特殊性，嵌入式软件在运行的时候不要占用过多的内存空间，要是像在 Windows XP 中一样，一个软件运行时占用 100 多兆空间的话，肯定是无法运行的。
- 对 CPU 资源的占用：嵌入式系统的 CPU 频率和桌面 PC 相比大都很低。比如目前流行的 Nokia 6681 手机，其 CPU 主频是 200 多 MHz，而桌面 PC 系统的 CPU 早已进入到 GHz 的时代。因此运行在上面的嵌入式软件自然也不能占用过多的 CPU 资源，同时在软件的算法实现上也不能消耗大量的 CPU 资源。

1.1.3 嵌入式系统的分类

我们依据嵌入式系统是否作为独立单元进行工作、是否具备实时操作、是否具备网络连接功能等方面，把嵌入式系统分为下面 4 种类型。

1) 单机嵌入式系统。单机嵌入式系统是以非网络方式进行工作的。这种嵌入式系统的输入可以是一个按钮命令，可以是一个脉冲信号，也可以是来自前一个设备的指令信号等。而它的输出可以是用户显示信息，也可以是一个电信号。通常情况下，单机嵌入式系统被应用在生产线流水作业中。

2) 网络设备。和单机嵌入式系统相对的是有一些嵌入式设备是和网络连接在一起的。网络可以是公司内部的局域网，也可以是 Internet 互联网。这些系统能够监测一些指定的参数，同时会通过网络把这些监测到的参数信息（如当地的风速、空气的温度、大气的压力、空气的湿度等信息）传送到远方的指挥中心或者信息处理中心，以便实现实时在线监控。

3) 实时的嵌入式系统。在上面介绍的网络设备中，有一些嵌入式系统可以通过网络把采集来的数据及时地送回到中央数据处理区，以方便随时对数据进行采集和监控。在这里就涉及到了另外一种嵌入式系统——实时嵌入式系统。这种系统要求设备在特定的时间内完成指定的任务。比如可以规定当某一地点的压力高于一定数值的时候，就自动打开阀门进行降压。

4) 移动型嵌入式系统。无论是网络设备还是实时嵌入式系统，大多依靠有线网络来传输数据。随着无线技术的迅速发展，以及无线网络设备的广泛应用，越来越多的嵌入式系统开始使用无线方式进行数据传输。移动型嵌入式设备可以随时随地地实现互联网接入、收发电子邮件等功能，大大扩展了嵌入式系统的应用领域和应用范围。

许多移动型嵌入式系统（设备）同时还具备应用程序下载功能，可以根据不同的应用场合

来选择相应的应用软件，极大地提高了嵌入式系统的通用性，在一定程度上也降低了嵌入式设备的使用成本。

1.2 嵌入式系统的组成

嵌入式系统一般由处理器、存储器、I/O 设备和相应的软件产品所组成。这里我们主要介绍嵌入式系统的处理器、存储器等和硬件相关的内容。首先来看嵌入式系统的处理器。

1.2.1 嵌入式系统处理器的分类

在嵌入式系统领域，嵌入式处理器的内核几乎全是 RISC 指令集的内核（RISC 的全称是 Reduced Instruction Set Computer，即精简指令集计算机）。RISC 是在 1979 年由美国加州大学伯克利分校提出的，它的体系结构具有如下一些特点：

- 采用固定长度的指令格式。
- 基本寻址方式有 2~3 种。
- 使用单周期指令，便于流水线操作执行。
- 大量使用寄存器，数据的处理指令只针对寄存器进行，有效地提高了指令的执行效率。

RISC 结构体系有两大主流：硅谷图形公司的 MIPS 技术和 ARM 公司的 Advanced RISC Machines 技术。从整个微处理器领域来讲，RISC 和 CISC（复杂指令集计算机）是目前设计制造微处理器的两种典型技术。虽然它们都是试图在体系结构、操作运行、软件硬件和编译时间等诸多因素中做出某种平衡，以求达到高效的目的，但由于各自采用的方法不同，因此在很多方面差异很大，它们的主要区别体现在以下 7 个方面：

1) 指令系统：RISC 的设计者把主要精力放在经常使用的指令上，尽量使它们具有简单高效的特色。对于不常用的功能，常通过组合指令来完成。因此，在 RISC 机器上实现特殊功能时，效率可能较低。但我们可以利用流水技术和超标量技术来加以改进和弥补。而 CISC 计算机的指令系统比较丰富，有专用指令来完成特定的功能。因此在处理特殊任务的时候，效率较高。

2) 存储器操作：RISC 对存储器的操作是有限制的；而 CISC 由于机器存储器的操作指令较多，因此操作较为直接。

3) 中断：RISC 机器在指令执行的适当地方可以响应中断；而 CISC 机器在指令执行结束后才可以响应中断。

4) CPU：RISC 的 CPU 包含较少的电路单元，因而面积小、功耗低；而 CISC 的 CPU 包含丰富的电路单元，因而功能强、面积大、功耗大。

5) 设计周期：RISC 微处理器结构简单，布局紧凑，设计周期短，易于采用最新技术；CISC 微处理器结构复杂，设计周期长。

6) 程序：RISC 汇编语言程序一般需要较大的内存空间，实现特殊功能时程序复杂；而 CISC 汇编语言程序编程相对简单，在科学计算和复杂操作的程序设计上相对容易。

7) 应用范围：由于 RISC 指令系统与特定的应用领域有关，故 RISC 机器更适合于专用