



Windows CE 项目开发实践丛书

Windows CE

工程实践完全解析

李大为 编著



中国电力出版社
www.cepp.com.cn

内容提要



Windows CE 项目开发实践丛书

Windows CE

工程实践完全解析

李大为 编著

1. Windows CE 工程实践完全解析 / 李大为编著. —北京: 中国电力出版社, 2008

中国版本图书馆CIP数据核字(2008)第111111号

责任编辑: 崔洪波
责任校对: 崔洪波
责任印制: 魏宇清

书 名: Windows CE
编 者: 李大为
出版社: 中国电力出版社
地址: 北京

电话: (010) 61879664

印 制: 北京中印联印务有限公司

开本尺寸: 185mm x 260mm

书 号: ISBN 978-7-5083-7473-4

版 次: 2008年10月北京第1版

印 次: 2008年10月第1次印刷

印 数: 0001—3000册

定 价: 45.00元

千字 616 字 85 印 张: 28

告 白 章

本 社 刊 物 均 有 邮 政 发 行 证 号 加 邮 局 发 行 函 购 均 可 直 接 向 本 社 定 购

本 社 刊 物 均 有 邮 政 发 行 证 号 加 邮 局 发 行 函 购 均 可 直 接 向 本 社 定 购



中国电力出版社

www.cepp.com.cn

内 容 提 要

本书是《Windows CE 项目开发实践丛书》之一。

本书讲述了 Windows CE 操作系统底层驱动软件开发技术。全书共分 5 篇 25 章, 内容包括开发 Windows CE 的 BootLoader、OAL、显示设备驱动程序、串口驱动程序和 PC 卡总线接口驱动程序, 通过解析和研读微软的 DeviceEmulator 开发板的 BSP 的源代码, 旨在帮助读者系统、全面、具体地了解 Windows Embedded CE 的 BSP 的原理与开发技术, 从而达到理解和实践 Windows CE 嵌入式系统软件开发目的。

本书注重系统性、先进性和实用性, 具有特色鲜明、逻辑性强、针对性强的特点。本书适合于从事 Windows CE 操作系统软件开发的工程技术人员使用, 也可作为高等院校自动化、电子、计算机、通信工程等专业师生的科研参考书。

图书在版编目 (CIP) 数据

Windows CE 工程实践完全解析 / 李大为编著. —北京: 中国电力出版社, 2008
(Windows CE 项目开发实践丛书)
ISBN 978-7-5083-7473-4

I. W… II. 李… 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 (2008) 第 136179 号

责任编辑: 崔素媛
责任校对: 崔燕菊
责任印制: 郭华清

书 名: Windows CE 工程实践完全解析

编 著: 李大为

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 68362602 传真: (010) 68316497

印 刷: 北京市同江印刷厂印刷

开本尺寸: 185mm × 260mm 印 张: 26 字 数: 646 千字

书 号: ISBN 978-7-5083-7473-4

版 次: 2008 年 10 月北京第 1 版

印 次: 2008 年 10 月第 1 次印刷

印 数: 0001—3000 册

定 价: 45.00 元

敬 告 读 者

本书封面贴有防伪标签, 加热后中心图案消失

本书如有印装质量问题, 我社发行部负责退换

版 权 专 有 翻 印 必 究

前言

我为什么写这本书

2006年的11月份，微软正式发布了 Windows CE 的 6.0 版本，全名为 Windows Embedded CE 6.0。伴随 Windows CE 6.0 版本的操作系统开发工具一起，微软还公开了它的全部内核态的源代码。对于有志于深入钻研 Windows CE 操作系统的技术人员来说，这无疑是一个特大的好消息。市面上充斥着大量的解析 Linux 内核源代码的书籍，却没有一本与之相应的解析 Windows CE 内核源代码的书，原因是显而易见的，因为这是微软首次向中国内地完全免费地公布 Windows CE 操作系统的内核源代码。

当时我正在微软的一家代理商那里做微软嵌入式 Windows 产品的技术支持工作，面对着这种有利的形势，当时我就萌生了写一本解析 Windows CE 内核源代码的书，内容应该包括：Windows CE 操作系统的进程（线程）调度机制、内存管理技术、设备驱动程序管理机制等。但是随后结合我所接触的 Windows CE 的使用者的情况来看，我发现他们中的大多数最需要的不是线程调度、内存管理之类的深入的技术内容，而是实实在在的 Windows CE 操作系统的底层移植。

由于嵌入式硬件的无标准、非规范的特性，嵌入式的操作系统都存在着一个底层移植的问题。厂商所提供的嵌入式操作系统一般都只是与特定嵌入式硬件平台无关的操作系统中间层，使用这些嵌入式操作系统开发具体的电子产品的开发者需要实现底层驱动程序以供操作系统的中间层使用。离开了这些底层的驱动程序，操作系统的中间层就无法操作硬件，于是整个操作系统的启动运行就无从谈起。微软将 Windows CE 的底层驱动程序使用标准的目录结构进行打包，并且命名为 BSP，意思是板级支持包（Board Support Package），这里所谓的“板”就是特定嵌入式硬件的电路设计主板，与之相应的，不同的嵌入式硬件设计就有不同的 BSP 软件包。由于为嵌入式操作系统开发底层驱动程序同时涉及操作系统的底层接口机制和对特定硬件操作的技术，这对开发者的技术能力提出了更高的要求，事实上对任何一款嵌入式操作系统开发底层驱动程序比开发上层应用软件的难度都要大得多。

大的嵌入式电子产品厂商通常以商业运作的手段解决技术所带来的问题。由于国家的法律不能够充分地保护知识产权，企业在技术方面的投资成果很容易随着技术人员的跳槽而被带走，甚至是带到企业的竞争对手那里。而国内的大多数企业管理水平都不够，不能够很有效地应对高科技产品的研发和技术积累的工作。此外，在中国的人才市场上获得合格的技术人才并不是很容易，企业的管理者们在有些情况下不得不迁就技术人才。所有这些都对国内的企业投资技术不利，技术的含量越高，投资的风险也就越大。于是，资金充裕的大厂商普

遍地选择用“钱”来解决技术，或者更确切地说是掌握技术的“人”所带来的各种问题。具体落实到 Windows CE 的使用者，底层驱动软件 BSP 来自芯片厂商，现在大的芯片厂商都开始做开发板级别的方案设计，芯片的使用者厂商只要肯出钱，完整的硬件设计方案以及服务于 Windows CE 操作系统的底层驱动软件都是可以获得的。Windows CE 的其余的开发工作还可以以微软的 TAM 服务（微软公司的一项收费的技术支持服务）为纽带同微软合作。留给国内的厂商的就只剩下单纯的产品集成和为数不多的硬件模块设计的任务，Windows CE 的开发活动变得更像是应用而不是研发，国内的厂商们自己的技术人员的作用被最大限度地弱化以适应国内高科技产品研发的大环境。

国内的厂商也许是不得已吧，但这样的开发模式却是掌握着芯片硬件和操作系统软件等关键上游技术的国外厂商所乐于见到的并且广泛推荐的。技术人才对于大厂商们这样的做法不能不有所了解，但是从纯粹技术的眼光来看，这样的技术开发模式却不具有代表性。对于国内的大多数小厂商来说，这样的开发模式也是不适用的，他们主要还是要依靠“人”而不是“钱”来完成 Windows CE 的技术开发任务。本书还是以技术的眼光来看待 Windows CE 的开发，鉴于以上所叙述的情况，最终我的这本书选择了以 Windows CE 的 BSP 开发技术为主要内容。

本书的内容介绍

尽管本书主要还是一本讲述 Windows CE 操作系统底层驱动软件开发的技术类书籍，但是 Windows CE 操作系统自有它不同于嵌入式 Linux 和 VxWorks 的特点，因而这本书也有一些独有的特色。Windows CE 相对于其他嵌入式操作系统在技术上的独特之处是它的一种与众不同的、我称之为“填空题”形式的开发模式，在绪论的《Windows CE 与软件工程》一节里对这种开发模式以及它的意义有详细的论述。Windows CE 的这种开发模式用简洁的语言可以这样来描述：微软确定软件的大框架；用户开发者补充完善具体细节，即所谓的“填空”。在这个开发模式之下，Windows CE 的底层驱动软件的开发者的工作不再是嵌入式 Linux 系统开发那样的个人英雄主义行为，而是一种标准化的、类似工厂生产线生产产品一样的企业经营活动。

除了技术层面上的独特的开发模式，Windows CE 操作系统的市场定位——消费电子使得这种开发模式有了极为强大的实用意义。由于只面向消费电子类的嵌入式产品，Windows CE 只支持 ARM、x86、MIPS 和 SH 4 种架构类型的 CPU，并且 Windows CE 对嵌入式外设的支持也不是像嵌入式 Linux 一样的无所不包、无所不能，而是重点支持消费电子类的嵌入式产品常用的外设。由于有了这样的硬件限定，所以尽管微软不能像对待桌面型的 Windows 操作系统那样包揽几乎所有的底层开发工作，却能够做比开发者所预想的要多得多的工作。这样做的好处是，一方面可以最大限度地减小 Windows CE 用户开发者的工作量；另一方面软件开发活动有了一定的标准将有利于企业对产品研发工作的管理。但是这种“填空题”形式的开发模式同时也使得微软的代码与用户自己开发的代码之间的接口变得复杂起来，这就好比当填空题的规模和复杂度达到一定的水平时，做填空题最棘手的不再是如何“填空”，而是“读懂题意”。对于 Windows CE 的使用者厂商来说，这种开发模式如果撇开它给企业管理带来的好处，单纯从产品研发的角度看，这其实是把开发成本转变为学习成本：Windows CE 的开发成本降低了，学习如何用 Windows CE 开发产品的难度却加大了。

本书的内容就是为了适应 Windows CE 的这种独特的填空题形式的开发模式，系统地帮

助 Windows CE 的开发者读懂“填空题”的题意。市面上也有一些 Windows CE 的书籍，它们往往一谈到 Windows CE 的设备驱动程序，就大讲设备管理器和流式接口设备驱动程序，似乎可以用一个模子把所有的 Windows CE 设备驱动程序都套进去。这显然是受到了嵌入式 Linux 的传统的、随意型的软件开发方式的影响，这样的内容对 Windows CE 的开发者当然也不能说全无意义，但是我认为这还远远不够，至少是没有真正领会 Windows CE 开发模式的精髓。本书的内容虽然不是像我原先预想的解析 Windows CE 操作系统的内核态源代码，却可以借鉴市面上的解析 Linux 内核源代码的书籍的写作方式，以 DeviceEmulator 的 BSP 为实例，深入地、系统地解析一个完整的 Windows CE 底层驱动软件的源代码。读者需要重点掌握以下两个方面的内容：哪些工作是用户必须做的，不说的话 Windows CE 操作系统就不能正常启动，或者不能按要求正确地运行；哪些工作是不需要用户来做的，而是可以直接利用微软的公共代码，这样的工作如果用户再做那就是“重新发明轮子”。

DeviceEmulator 是微软设计的一款以三星 S3C2410 CPU 为基础的虚拟硬件开发板，伴随 Windows CE 6.0 的开发工具 Platform Builder 软件，微软向用户提供了它的完整的 BSP 源代码和虚拟运行环境。“虚拟”指的是这个硬件平台以及它的运行环境都不是实实在在的物理硬件，而是通过软件来模拟嵌入式硬件在用户的开发 PC 上运行，类似 VMware 和 Virtual PC 等软件模拟标准 PC 的运行。尽管软件是虚拟的，但是 DeviceEmulator 的前身是基于 Windows CE 5.0 及以前版本的一款真实的三星 S3C2410 开发板，用户完全不必担心它的有效性。由于 Windows CE 开发模式的独特性，微软推荐用户以一套现成的 BSP 软件为基础，在它的源代码之上进行增补修改以开发出适合自己的硬件平台的底层软件 BSP。微软为 Windows CE 所支持的 4 大类 CPU 分别提供了一个或者多个用作样板和开发起点的 BSP，DeviceEmulator 就是基于 ARM 架构的 CPU 的开发板的 BSP 开发样板之一。

我之所以选择以 DeviceEmulator 为范本介绍 Windows CE 的底层软件 BSP 的开发，是因为 S3C2410 CPU 和 DeviceEmulator 的板级硬件设计相对简单，读者可以将主要的注意力放在软件上，而不必过多地受到硬件问题的干扰。

全书共分为 5 篇：第一篇，开发 Windows CE 的 BootLoader；第二篇，开发 Windows CE 的 OAL；第三篇，开发 Windows CE 的显示设备驱动程序；第四篇，开发 Windows CE 的串口驱动程序；第五篇，开发 Windows CE 的 PC 卡总线接口驱动程序。

在 5 篇的基础上，本书正文划分为 25 章。第 1~4 章属于第一篇，讲述 Windows CE 的 BootLoader 的开发技术。第 1 章讲述 Windows CE 的 BootLoader 的理论基础知识。第 2 章介绍微软为 Windows CE 的 BootLoader 所设计的标准软件框架和支持库。第 3 章以 DeviceEmulator 的 BootLoader 为实例讲述需要 Windows CE 的用户负责开发实现的功能函数，包括必须实现的函数和可选实现的函数。第 4 章通过解析 DownloadImage 函数来认识 Windows CE 操作系统的镜像文件的格式，DownloadImage 函数的功能是负责解析 BootLoader 通过下载或者本地获取所得的 Windows CE 操作系统镜像文件，它本来是一个不需用户开发实现的 BLCOMMON 支持库函数。解析这个函数的目的是初步认识 Windows CE 的镜像文件的格式，这对 Windows CE 开发和调试的工作也许会有所帮助。

第 5~11 章属于第二篇，讲述 Windows CE 的 OAL 的开发技术。第 5 章讲述 Windows CE 操作系统的启动执行流程，它实质上就是用户开发实现的 OAL 函数所服务的软件大框架。第 6 章是需要 Windows CE 的用户开发实现的 OAL 函数的汇总。第 7 章讲述 Windows CE 操

作系统的中断处理机制。第 8 章讲述 Windows CE 操作系统的系统时钟管理技术。第 9 章介绍 Windows CE 的 TOC, 这可以看作是第 4 章的延续和深入, 其内容属于较为深入的认识 Windows CE 操作系统镜像文件的格式。第 10 章讲述 Windows CE 的 KITL 调试技术, 具体地说就是用户为了启用 Windows CE 的 KITL 功能需要做哪些工作。KITL 连接功能对于 Windows CE 的调试工作意义十分重大, Platform Builder 中的 Remote 系列的调试功能都要以目标设备与开发 PC 之间的 KITL 连接为基础。第 11 章以 CS8900A 芯片为实例介绍 Windows CE 以太网端口设备 EthDbg 驱动程序的开发技术。Windows CE 的 BootLoader 普遍会采用以太网作为下载操作系统镜像的端口, 而且 KITL 连接最常见的也是通过以太网, 因而以太网接口的 EthDbg 驱动程序在 BootLoader 和 KITL 中普遍都会被使用。

第 12~15 章属于第三篇, 讲述 Windows CE 的显示设备驱动程序的开发技术。显示设备的驱动程序在 Windows CE 的设备驱动程序中比较特殊, 它不是由设备管理器 device.exe, 而是由图形窗口与事件子系统 gwes.exe 负责加载和管理。第 12 章讲述 Windows CE 的 DDI 函数集合。DDI 系列函数由特定的显示设备驱动程序动态链接库向 gwes.exe 导出, 它们本来不需要用户开发实现, 但是它们的实现代码参与了构造特定的显示设备驱动程序动态链接库。可以把 DDI 函数集合看做是 Windows CE 的显示设备驱动程序的 MDD 层, 了解它是为了更好的掌握显示设备驱动程序的 PDD 层。第 13 章讲述 Windows CE 的显示表面、调色板、剪切板等概念, Windows CE 系统使用这些概念及其技术管理图形用户界面。第 14 章介绍 Windows CE 的 GPE 类。GPE 类是开发 Windows CE 的用户开发实现显示设备驱动程序 PDD 层工作的起点, 所有用户特定的显示设备驱动程序 PDD 层都以 GPE 类 (或者它的继承类) 为父类。第 15 章讲述 Windows CE 的显示设备驱动程序对 DirectDraw 的支持。DirectDraw 是对图形图像显示进行二维加速的支持功能, 凡是需要支持 DirectDraw 功能的用户特定显示设备驱动程序的 PDD 层都应该以 DDGPE 类为父类, DDGPE 类是 GPE 类的继承类。

第 16~20 章属于第四篇, 讲述 Windows CE 的串口驱动程序的开发技术。第 16 章讲述 Windows CE 的设备管理器和受设备管理器加载管理的流式接口设备驱动程序的基础理论知识, 是本篇后续各章的知识总括。第 17 章重点解说 Windows CE 的串口驱动程序向设备管理器所导出的代表串口设备句柄的 HW_INDEP_INFO 结构体数据, 串口驱动程序的各个对外导出接口函数都以这个 HW_INDEP_INFO 结构体的数据作为操作对象, 掌握了这个结构体的所有各个成员的含义就等于掌握了串口驱动程序的 MDD 层。第 18 章介绍 CSerialPDD 类, 这个类是 Windows CE 的串口驱动程序 PDD 层开发工作的基础, 它可以与显示设备驱动程序的 GPE 类对比, 所有用户的特定串口驱动程序的 PDD 层都以 CSerialPDD 类 (或者它的继承类) 为父类。第 19 章介绍 Windows CE 操作系统对外设的电源管理技术。Windows CE 所面向的消费电子产品多以电池为供电电源, 对外设的电源管理技术以节电省电为首要目标。第 20 章以 DeviceEmulator 的 S3C2410 芯片内置的 UART 端口的驱动程序为实例具体解说 Windows CE 串口驱动程序的开发技术。

第 21~25 章属于第五篇, 讲述 Windows CE 的 PC 卡总线接口驱动程序的开发技术。第 21 章讲述 PC 卡总线接口驱动程序的基础理论知识。PC 卡总线接口驱动程序本质上仍属于流式接口驱动程序, 受设备管理器加载和管理, 但是它的分层结构却比较复杂, 本章的目的就是理清这个复杂的分层结构。第 22 章介绍 CPCCardBusBridgeBase 类, 这个类是用户开发 Windows CE 的 PC 卡总线桥驱动程序工作的起点, 它的作用和原理可以与显示设备驱动程序

中的 GPE 类和串口驱动程序中的 CSerialPDD 类进行类比掌握。第 23 章解说 DeviceEmulator 的 PC 卡总线桥驱动程序具体开发实现的实例 CPcmciaBusBridge 类,它是 CPCCardBusBridge-Base 类的继承类。第 24 章介绍 Windows CE 的 PC 卡总线桥驱动程序中的 Socket 的概念。第 25 章介绍 PC 卡总线桥驱动程序中的 ISR 和 IST。

致 谢

首先也是最重要的是感谢我的父亲和母亲,他们不仅支持我写书,而且在我辞去工作专心写书的期间,还要依靠他们照料我的生活。我对他们的感念之情,不是一个“谢”字可以表达得了的!

其次感谢研华(中国)公司和微软公司,感谢他们给了我一个直接接触微软嵌入式 Windows 技术的机会,在此谨向他们表示我诚挚的谢意!

感谢我的女朋友王萍,她不仅全力支持我写这本书,而且还在此过程中提供了不少有价值的建议。

我的朋友俞孝松、孙志新、黄开元等,在这本书的写作过程中,我经常同他们进行交流与讨论,书中有多处内容直接得益于他们的宝贵意见与帮助,在此表示郑重感谢。另外,本书的编写还得到了崔旖、曹跃隆、王旭等朋友的支持和帮助,在此一并致谢。

编 者

2008 年 5 月

目 录

前 言	1
绪 论	1

第一篇 开发 Windows CE 的 BootLoader

第 1 章 BootLoader 的理论知识介绍	16
1.1 BootLoader 是大多数嵌入式操作系统运行最初的开始	16
1.2 微软建议 Windows CE 的 BootLoader 所应实现的功能	16
1.3 BootLoader 并不是基于 Windows CE 的嵌入式软件系统的必不可少的组成部分	17
第 2 章 Windows CE BootLoader 的软件框架和支持库	19
2.1 以 Eboot 为实例学习 Windows CE BootLoader 的软件框架	19
2.2 BLCOMMON 支持库	21
2.3 Eboot 支持库	21
2.4 EthDbg 驱动程序	24
第 3 章 案例解说开发实现 OEM 函数	27
3.1 DeviceEmulator 虚拟平台的硬件设计	27
3.2 开天辟地的 StartUp 函数	29
3.3 控制流 OEM 函数	34
3.4 调试功能 OEM 函数	37
3.5 下载功能 OEM 函数	39
3.6 以太网收发帧 OEM 函数	40
3.7 Flash 操作 OEM 函数	41
3.8 时钟功能 OEM 函数	43
3.9 可选实现的 OEM 函数	44
3.10 在 BootLoader 和操作系统之间共享信息的数据结构	46
3.11 保存用户选项配置参数的数据结构	49
第 4 章 认识 Windows CE 操作系统镜像文件的格式	52

第二篇 开发 Windows CE 的 OAL

第 5 章 Windows CE 操作系统的系统启动执行流程	58
5.1 图览 Windows CE 的初始化启动流程	58

5.2	开天辟地的 StartUp 函数	59
5.3	KernelStart 函数	60
5.4	ARMInit 函数	64
5.5	kernel.dll 模块的入口点函数 NKStartup	64
5.6	ARMSetup 函数	66
5.7	KernelStart 函数	69
第 6 章	案例解说 Windows CE OAL 的开发任务汇总	70
6.1	OAL 是什么	70
6.2	OEMGLOBAL 结构体及其全局变量	70
6.3	OEM 地址映射表	78
6.4	OEMIoControl 函数	79
6.5	OAL 的 Cache 操作 OEM 函数	80
第 7 章	Windows CE 操作系统的中断处理机制	82
7.1	ISR 和 IST 协同处理外设的中断请求	82
7.2	设置系统中断向量表	82
7.3	初始化函数 OEMInit 并启用硬件中断处理机制	85
7.4	实现 OEMInterruptHandler 函数供操作系统处理 IRQ 中断	88
7.5	实现 OEMInterruptHandlerFIQ 函数处理 FIQ 中断	93
7.6	案例解说中断处理的另一组成部分 IST	94
7.7	为 IST 做些什么事情	101
第 8 章	系统时钟管理	105
8.1	初始化函数 OEMInit 并启用系统时钟	105
8.2	ISR 必须处理系统时钟的定时中断请求	111
第 9 章	TOC 的秘密	114
9.1	初识 pTOC 指针	114
9.2	解析实例以加深认识 TOC 数据	116
9.3	TOC 的条目	120
9.4	TOC 数据的实质	121
第 10 章	KITL 调试技术	123
10.1	什么是 KITL	123
10.2	图览 KITL 的初始化流程	123
10.3	初始化 KITL 库	125
10.4	初始化 KITL 连接	127
10.5	OAL_KITL_ARGS 结构体	130
10.6	OAL_KITL_DEVICE 结构体	133
10.7	主动的 KITL 模式与被动的 KITL 模式	135

10.8	KITLTRANSPORT 结构体	137
10.9	汇总 OAL 的开发者需要为启用 KITL 功能做哪些事情	140
第 11 章	案例解说以太网端口设备 EthDbg 驱动程序的开发	143
11.1	CS8900A 的硬件原理知识	143
11.2	读写 ReadPacketPage 和 WritePacketPage	145
11.3	硬件初始化函数 CS8900AInit	147
11.4	发送以太网数据帧函数 CS8900ASendFrame	150
11.5	接收以太网数据帧函数 CS8900AGetFrame	151
11.6	启用与禁用 CS8900A 的中断功能的函数 CS8900AEnableInts 和 CS8900ADisableInts	152
11.7	CS8900A 的配置地址过滤机制的函数 CS8900ACurrentPacketFilter 和 CS8900AMulticastList	153
第三篇 开发 Windows CE 的显示设备驱动程序		
第 12 章	从 DDI 系列函数开始	158
12.1	DDI 函数是什么	158
12.2	Windows CE 的 GWES 子系统如何获得 DDI 函数	158
12.3	DrvGradientFill 和 DrvAlphaBlend 函数	164
12.4	DrvEnablePDEV 和 DrvDisablePDEV 函数	166
12.5	DrvEnableSurface 和 DrvDisableSurface 函数	174
12.6	DrvCreateDeviceBitmap 和 DrvDeleteDeviceBitmap 函数	175
12.7	DrvRealizeBrush 函数	177
12.8	DrvStrokePath 函数	178
12.9	DrvFillPath 函数	179
12.10	DrvPaint 函数	182
12.11	块传输功能函数 DrvAnyBlt、DrvBitBlt、DrvCopyBits 和 DrvTransparentBlt	182
12.12	DrvSetPalette 函数	187
12.13	DrvSetPointerShape 函数	187
12.14	DrvMovePointer 函数	188
12.15	DrvGetModes 函数	189
12.16	DrvRealizeColor 和 DrvUnrealizeColor 函数	190
12.17	DrvGetMasks 函数	190
12.18	DrvContrastControl 函数	191
12.19	DrvPowerHandler 函数	192
12.20	DrvEscape 函数	192
12.21	DrvDisableDriver 函数	193
第 13 章	Windows CE 的显示表面及与之相关的数据结构	194
13.1	显示表面	194
13.2	调色板、颜色转换对象和颜色转换类	199

13.3	SURFOBJ 结构体和 TmpGPESurf 类	204
13.4	路径	205
13.5	剪切板对象 CLIPOBJ	207
13.6	画笔	210
第 14 章	GPE 类是用户开发 Windows CE 的显示设备驱动程序工作的起点	211
14.1	从 DDI 到 GPE	211
14.2	GPE 类	212
14.3	GPE 类和块传输操作	217
14.4	GPE 类和画线操作	220
14.5	DeviceEmulator 的显示设备驱动程序实现——S3C2410DISP 类	223
第 15 章	显示设备及其驱动程序对 DirectDraw 功能的支持	233
15.1	由导出函数 HALInit 引出 DDHALINFO 结构体	233
15.2	DDHALINFO 结构体	234
15.3	DDCAPS 结构体	236
15.4	DDHAL_DDCALLBACKS 结构体	243
15.5	DDHAL_DDSSURFACECALLBACKS 结构体	244
15.6	DirectDraw 显示表面	245
15.7	DDGPE 类	249
第四篇 开发 Windows CE 的串口驱动程序		
第 16 章	设备管理器和流式接口设备驱动程序	254
16.1	串口驱动程序是典型的流式接口设备驱动程序	254
16.2	设备管理器以总线枚举的方式加载流式接口的设备驱动程序	256
16.3	总线枚举器加载一个设备驱动程序的具体步骤	259
第 17 章	串口设备句柄和串口文件句柄	262
17.1	从 COM_Init 导出函数到 HW_INDEP_INFO 结构体	262
17.2	pHWObj 成员	264
17.3	pHWHead 成员	268
17.4	DCB 成员	268
17.5	CommTimeouts 成员	273
17.6	RxBufferInfo 和 TxBufferInfo 成员	275
17.7	pAccessOwner、OpenList、OpenCS 和 OpenCnt 成员	276
17.8	fEventMask 成员	280
17.9	pDispatchThread、Priority256、hKillDispatchThread、KillRxThread 和 hSerialEvent 成员	280
17.10	hReadEvent 和 hTransmitEvent 成员	281
17.11	fAbortRead 和 fAbortTransmit 成员	282

17.12	DroppedBytesMDD、DroppedBytesPDD、RxBytes、TxBytes、TxBytesPending 和 TxBytesSent 成员	282
17.13	XFlow、StopXmit、SentXoff、DtrFlow 和 RtsFlow 成员	283
17.14	ReceiveCritSec1 和 TransmitCritSec1 成员	284
17.15	dwMddErrors 成员	284
第 18 章	CSerialPDD 类是 OEM 用户开发 Windows CE 串口驱动程序 PDD 层工作的起点	286
18.1	CSerialPDD 类定义	286
18.2	CSerialPDD 类的初始化部分	289
18.3	CSerialPDD 类的串口设备操作部分	290
18.4	CSerialPDD 类的中断接口部分	291
18.5	CSerialPDD 类的发送与接收部分	292
18.6	CSerialPDD 类的 MODEM 功能部分	293
18.7	CSerialPDD 类的线路功能部分	294
18.8	CSerialPDD 类的串口配置部分	295
18.9	CSerialPDD 类的 IR 特殊处理部分和错误处理部分	299
第 19 章	Windows CE 操作系统对外设的电源状态管理	300
19.1	电源管理架构和电源管理器	300
19.2	设备电源状态和系统电源状态	301
19.3	电源管理器与设备驱动程序之间的编程接口	305
19.4	电源管理器与应用程序之间的编程接口	309
19.5	例说设备驱动程序开发中的电源管理实务	310
第 20 章	案例解说串口驱动程序的具体开发	314
20.1	DeviceEmulator 的串口驱动程序 PDD 层的类定义	314
20.2	CReg2410Uart 类	316
20.3	S3C2410 串口驱动程序 PDD 层的初始化	321
20.4	CPdd2410Uart 类的 PostInit 成员函数	326
20.5	串口数据接收中断请求处理函数 ReceiveInterruptHandler	329
20.6	串口 MODEM 事件中断请求处理函数 ModemInterruptHandler	331
20.7	串口线路中断请求处理函数 LineInterruptHandler	331
20.8	串口驱动程序 PDD 层的流控功能函数 XmitComChar、SetDTR 和 SetRTS	332
20.9	串口数据发送中断请求处理函数 XmitInterruptHandler	333
20.10	串口发送中断信号的功能函数 SetBreak	334
20.11	CancelReceive 函数和 CancelXmit 函数	335
20.12	SetOutputMode 函数和 SetBaudRate 函数	336

第五篇 开发 Windows CE 的 PC 卡总线接口驱动程序

第 21 章	PC 卡总线桥驱动程序	338
21.1	PC 卡驱动程序的分层结构	338

21.2	PC 卡接口驱动程序实体分离的 MDD 层和 PDD 层	340
21.3	PC 卡总线桥驱动程序	341
第 22 章	CPCCardBusBridgeBase 类是 OEM 开发者开发 PC 卡总线桥驱动程序工作的起点	343
22.1	CPCCardBusBridgeBase 类定义	343
22.2	CPcmciaWindowInfo 类	344
22.3	CPCCardBusBridgeBase 类的数据成员和成员函数	348
第 23 章	CPcmciaBusBridge 类——DeviceEmulator 的 PC 卡总线桥驱动程序具体实现	354
23.1	CPcmciaBusBridge 类的抽象意义及继承关系	354
23.2	CPcmciaBusBridge 类的数据成员	355
23.3	CPcmciaBusBridge 类的构造函数要负责生成 PC 卡总线桥驱动程序的 IST 线程	361
23.4	CPcmciaBusBridge 类的 Init 成员函数负责初始化 PC 卡总线桥硬件和向上提交 Socket Service	363
第 24 章	PC 卡的 Socket	370
24.1	Socket Service 的内容	370
24.2	DeviceEmulator 的 PC 卡 Socket 的类数据结构及其继承关系	372
24.3	Socket 的属性（信息）和状态	377
24.4	PC 卡 Socket 在系统中的组织管理	381
24.5	Socket 的句柄	385
24.6	Socket 的电源管理	386
24.7	PC 卡总线桥设备与 PC 卡 Socket 在软件数据结构上的关联	389
第 25 章	PCMCIA 桥驱动程序的 IST 线程和 ISR	393
25.1	CPcmciaBusBridge 类成员函数 InstallIsr 初始化 PCMCIA 控制器的卡状态改变中断	393
25.2	PC 卡总线桥驱动程序的 IST 线程负责处理 PC 卡插拔和 Socket 电源状态改变事件	395
参考文献		399

绪 论

什么是 Windows CE

1. 从纯粹软件技术的角度看 Windows CE

从纯粹软件技术的角度看 Windows CE, 则可以说 Windows CE 就是嵌入式的 Windows, 虽然 Windows CE 不是微软唯一的一款嵌入式操作系统产品, 但无疑是最有代表性的一款。在纯粹技术的方面, Windows CE 和嵌入式的 Linux、VxWorks 等种类繁多的嵌入式操作系统没有本质区别。操作系统大家都不陌生, 最常见的供个人办公和娱乐用的个人计算机没有了操作系统是无法使用的, 很多非技术的白领人士甚至常常把计算机和 Windows 操作系统等同起来, 他们不能想象离开了蓝天白云桌面的个人电脑是个什么样子。除了将计算机与操作系统混为一谈外, 我所见过的也有不少甚至是技术的从业人员将操作系统与(安装)光盘等同起来, 因为桌面版的操作系统, 无论是 Windows 还是 Linux, 通常情况下都是以光盘为安装介质的。Windows CE 是嵌入式的而不是桌面型的操作系统, 对待嵌入式的操作系统绝不能犯以上两种混淆的错误。

包括 Windows CE 在内的所有类型的嵌入式操作系统, 其区别于桌面操作系统的本质特点, 一是可裁减, 二是可移植。嵌入式操作系统的这两个本质特点是由运行嵌入式操作系统的嵌入式硬件的特点决定的。嵌入式的硬件平台首先是由于成本的限制造成其硬件资源较为稀缺。以掌上型的消费电子设备 PDA 为例, 用技术的眼光看, 它也是一个完整的计算机系统, 所谓“麻雀虽小, 五脏俱全”。但是在这样的计算机系统中绝没有个人计算机上的那种动辄几百 MB 乃至 GB 的物理内存和几百 GB 容量的辅助存储设备, 目前一般只有几十 MB 的 RAM 内存和几十 MB 的 Flash 存储器, 它的 CPU 的运行速率与主流配置的个人计算机相比也要相差一个数量级。以这样的硬件平台为基础的嵌入式操作系统对硬件资源的使用就不得不精打细算。针对嵌入式的电子产品应用功能比较专门化的特点, 比如 MP4 主要的功能是音视频媒体播放, 而 GPS 就是定位导航, 嵌入式操作系统实现不同功能的各个组成部分最好要能做到按需选用, 即特定的产品需要的功能就留下, 不需要则不留, 这就是可裁减。经过裁减的嵌入式操作系统不仅意味着较少地占用存储空间, 而且由于软件的运行负荷较小, 对于目标嵌入式系统的启动速度和运行速度也会大有好处。

针对以 Windows CE 为代表的嵌入式操作系统的可裁减性, 有人做过这样一个形象的比喻: 如果把操作系统比作是一套住房, 那么桌面版的 Windows 用户就相当于买了一套现成的商品房, 里面的房间、装修和家具一应俱全, 而且这套房子与同一楼盘里的其他商品房在设计格局与内部的配套物品上都是完全相同的; 而 Windows CE 的用户则好像是买到了一堆砖块、水泥、沙子等建筑材料, 他们可以使用这些材料盖一栋摩天大楼, 也可以搭一个小小的

狗窝，完全取决于用户需要的是摩天大楼还是狗窝。

嵌入式操作系统的另一特性——可移植性——源于嵌入式硬件平台的非标准化。桌面版的操作系统所面向的个人计算机是有一个世界统一的硬件标准的，通常称为 PC 标准。符合 PC 标准的计算机系统，其 CPU 必定是 x86 架构的，其主板上必定少不了南桥和北桥，连接南桥和北桥的必定是 PCI 总线，甚至串口和并口等外设的 I/O 端口地址都是固定有序的等。在这个硬件标准之下，世界上任何一个人计算机生产厂家的产品在技术原理上都是大同小异的，只在质量和高低端配置上有所区别。个人计算机由于有了这么一个世界统一的硬件标准，运行于其上的桌面版 Windows 操作系统不存在移植的问题，微软做了所有的事情，用户最多只需增加少量的满足特定应用需求的应用软件和不常见的外设编写驱动程序。微软将这个运行在世界统一标准的硬件之上的全世界统一版本的 Windows 操作系统制作成标准的安装光盘，使用者只需将它置入光驱中然后一路单击“下一步”按钮就可以完成安装然后使用。

相比之下，嵌入式的操作系统就没有这样的好运气。嵌入式的硬件平台没有类似 PC 标准的世界统一硬件标准，它的 CPU 可能是 ARM 的，或者 MIPS 的，当然也可能是 x86 的，CPU 之外的外设使用情况也是千差万别，毫无标准可言。在这种情况下，任何一款嵌入式操作系统的开发厂商和开发者都无法事先设计和实现适合所有嵌入式硬件平台的操作系统，他们只能做与硬件无关的事情，或者虽与硬件有关，但却是大多数的硬件平台都需要的带有普遍共性的软件内容，通常称为操作系统中间层。这就给嵌入式操作系统的使用者们带来了麻烦，因为原厂商和开发者没做的事情就需要他们自己去做，具体地说，使用者需要为 Windows CE 操作系统开发底层软件。底层软件是由微软设计开发的与硬件无关的操作系统中间层和特定的目标嵌入式硬件平台之间的接口，它们具体负责操作各不相同的嵌入式硬件，并且向操作系统的中间层提供具有微软预定义标准程序接口的初始化硬件和对硬件实现读写访问的功能。使用者然后将这些底层软件与微软的操作系统中间层，可能还要加上应用软件，三者一起经过 Windows CE 开发工具编译链接，最后打包成可以在目标嵌入式设备上运行的嵌入式操作系统镜像。这就是嵌入式操作系统的可移植特性。

嵌入式操作系统的可移植特性，与其说是“可”移植，还不如说是“必须”移植，因为不移植就无法使用。于是，Windows CE 的使用者们没有可以直接通过光盘安装的 Windows CE 操作系统。在我做技术支持期间，不止一个客户向我索要过 Windows CE 的安装光盘，我只能很遗憾地回答没有，不仅我没有，微软也没有，任何一家嵌入式操作系统的开发厂商都无法为客户提供这样的方便。

在我看来，真正的技术意义上的嵌入式操作系统必须同时具备以上两个最本质的特性：可裁减和可移植。至于其他的特点其实都是由这两个最根本的特点衍生而来的，比如嵌入式操作系统一般较少占用存储空间，这个特点就是由它们的可裁减性带来的。此外，嵌入式操作系统占用存储空间的大小的总趋势也是随着嵌入式硬件资源紧张程度的不断缓解而不断增大的，所以较少占用存储空间的特点只是嵌入式操作系统的一个一般的特征，算不上是本质的特性。相比之下，桌面版的操作系统就不是可裁减和可移植的。桌面版的 Windows 操作系统所面向的个人计算机都是通用功能的，用户既可以拿它们做办公用，也可以看影碟或者玩游戏，并且硬盘和内存的存储容量都是足够充裕的，桌面版的操作系统既没有裁减的可行性也没有裁减的必要性。而使用光盘安装的 Windows 操作系统，运行在技术原理完全相同的硬件平台之上，当然也就没有移植性可言。

以上的文字详细说明了嵌入式的操作系统与桌面型操作系统之间的技术上的区别。但是更要准确、更充分地解释“什么是 Windows CE”，那就还需要说明 Windows CE 作为种类众多的嵌入式操作系统之一与其他类型的嵌入式操作系统之间的区别，最重要的是说明与其他嵌入式操作系统相比较 Windows CE 的优势在哪里。虽然前面说明了 Windows CE 与桌面版的 Windows 操作系统有那么大的区别，但是如果说 Windows CE 相对于其他嵌入式操作系统在技术方面的优势所在，则几乎可以一言以蔽之——Windows 操作系统在桌面操作系统中的技术优势其实就是 Windows CE 在嵌入式操作系统中的技术优势。下面以 Windows CE 与嵌入式的 Linux 作比较，分四个方面说明 Windows CE 操作系统的技术优势。

(1) Windows CE 可以保证操作系统的软件产品质量。

Windows CE 毕竟是由世界知名度最高的软件厂商设计开发的操作系统，质量比较有保证。相比之下，Linux 则主要是由一群组织松散的所谓“黑客”提供的程序软件。负责组织和管理工作 Linux 开源项目的 GNU 组织希望打破软件厂商对操作系统源代码的垄断，在他们看来这样做不利于软件技术的推广与交流，所以鼓励以开放源代码的形式开发和使用操作系统。这样的目标在我们所处的这个时代虽然显得有些不够务实，却也还算崇高。但是对于大多数的通过国际互联网联系起来的做基础工作的 Linux 开源项目的开发者们来说，除去纯粹的兴趣爱好以外，他们为 Linux 做事情的主要目的是出名。没有人向他们付费，他们也不用为软件的质量问题承担责任，于是重开发而轻测试就成了必然。此外，还有一个问题是由于组织松散以致各个开源项目之间缺乏协调一致，比如某个版本的 Linux 内核只能或者不能用某个或者某些版本的 GCC 工具进行编译，否则就会出错。类似这样的现象给 Linux 的使用者带来了不少的麻烦。这些问题对于版本众多、使用不够充分、对提高开发者的知名度又帮助较小的嵌入式版本的 Linux 操作系统尤其严重，我曾经亲身经历过，从 Linux 的官方网站下载的源代码，一字不改的情况下竟然编译不能通过。

(2) Windows CE 有大量的经过嵌入式处理的应用软件和操作系统特性可供用户灵活选用。

任何的计算机系统总是要依靠应用软件才能向最终用户提供使用价值，除用户自己或者第三方开发以外，桌面版操作系统的厂商都会随操作系统一起向用户提供一些常用的应用程序和操作系统特性 (feature)，前者主要包括网页浏览器、媒体播放器、办公软件等，后者则以网络协议栈、文件系统、多语言的支持库、音视频多媒体的编码解码器为代表。Windows CE 是嵌入式的操作系统，微软也随它一起向使用者提供了大量的这两种类型的软件，并且给它们起了一个更时髦、更有抽象意义的名字——用户体验 (user experiences)。操作系统的厂商无疑是想通过这些用户体验向用户提供开发使用时的最大便利从而吸引客户，应用软件可以代替用户在需要时自己开发或者花钱另买，而操作系统特性可以在用户自己或者第三方开发应用软件时提供强有力的支持。Windows CE 所附带的应用软件与在桌面 Windows 上的对应的软件相比，名字都是相同的，界面和功能也是相似的，并且都经过了嵌入式处理。所谓嵌入式处理，还是裁减的意思，但却不是供用户按需选用的裁减，而是在操作系统出厂之前的内容精简。以网页浏览器举例来说，桌面版的 Internet Explorer 6.0 占用存储空间超过 100MB，硬件资源有限的嵌入式系统当然不能接受。Windows CE 所带的网页浏览器也叫 Internet Explorer，操作界面与桌面版的 IE 也是相似的，但是它去除了桌面版 IE 中不常用的功能，只保留了最核心的功能，界面也大大地简化了，以达到降低对存储空间和 CPU 运行速度的要求的目的。