



Windows CE 项目开发实践丛书

Windows CE

设备驱动及BSP开发指南

周建设 编著



中国电力出版社

www.cepp.com.cn



Windows CE 项目开发实践丛书

Windows CE

设备驱动及BSP开发指南

周建设 编著



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。

本书围绕着一个 BSP 的实例开发为中心进行讲解,对 Windows CE 设备驱动程序的开发、调试、移植、Boot Loader 开发、OAL 开发、BSP 相关配置文件的设置等做了深入地剖析。此外,还在此实例 BSP 基础上介绍了开发家庭智能化系统的流程。

本书可作为高等院校电子、电气、控制、计算机等专业本科生、研究生学习 Windows CE 设备驱动及 BSP 开发的参考书或自学教材,也可供从事嵌入式领域的科研和工程技术人员参考使用,还可作为嵌入式培训班的教材。

图书在版编目 (CIP) 数据

Windows CE 设备驱动及BSP开发指南 / 周建设编著. —北京:
中国电力出版社, 2009

(Windows CE 项目开发实践丛书)

ISBN 978-7-5083-8632-4

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

中国版本图书馆CIP数据核字 (2009) 第043074号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.cepp.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2009 年 6 月第一版 2009 年 6 月北京第一次印刷

787 毫米 × 1092 毫米 16 开本 26 印张 634 千字

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

序



科技日新月异，Windows CE 自问世以来，已在市场上占有一席之地，也让嵌入式开发人员多了一个选择。Windows CE 的内核随着用户的需求而不断改进，开发的工具也更加友好，使得更多的开发者能够更简单、更快速地使用 Windows CE 强大的功能。

作者本身已有单片机开发的经验，进入西安昭营科技之后，学习 Windows CE 的开发，如鱼得水，他学习的热诚，让我印象深刻。后来指导公司新进研发人员，也非常热心。其间，在几次西安昭营科技的 Windows CE 研讨会做技术演讲。在公司 Windows CE 的技术支持上也颇有创新。

恰逢这次出版书籍的机会，我便鼓励他将 Windows CE 技术的心得整理分享给有心于此的开发者。希望他的分享能让更多的人进入 Windows CE 领域。

陈志源

2008 年 12 月 26 日于台北



前言

Windows CE 经过 12 年的发展, 现在已成为嵌入式领域的一颗巨星。Windows CE 具有的组件化、可定制性、与 Windows 平台兼容等特性, 在商业开发方面的优势早已表现出来了。作为工厂比较密集的中国已经逐渐意识到产品质量的重要性, 很多从事嵌入式产品生产的厂家已经开始放弃低成本、低品质、低收益、高产量的经营策略, 开始对 Windows CE 体系结构、服务以及技术支持产生了浓厚的兴趣, 并且在其产品比如智能手机、家庭自动化、信息家电、个人导航、工业控制等领域中都开始采用 Windows CE。

自 Windows CE 第六版 (Windows Embedded CE 6.0) 开始, 它的核心源码是 100% 共享的, 这极大地方便了 Windows CE 工程师在开发过程中对疑难问题的跟踪调试, 也可以参考其中的代码来开发自己的产品, 但它不需要遵循 GPL 协议。

由于使用 Windows CE 开发产品具有强大的优势, 因此越来越多的公司选择使用 Windows CE 进行开发。但是 Windows CE 的设备驱动以及 BSP 开发对一般的用户来说还是一个难点, 虽然微软提供了多种设备驱动程序以及 BSP 的样例, 但是对许多初学者来说根本不知道如何入手学习 Windows CE 设备驱动及 BSP 开发, 本书以实用、简洁为宗旨, 旨在给读者提供一个纲领性的参考, 希望读者能快速地掌握 Windows CE 设备驱动及 BSP 开发的相关方法, 并且经过适当的训练之后就可以独立地开发出项目中需要的设备驱动或 BSP。

Windows CE 的内核发展

在 Windows CE 12 年的发展历史中, 出现过三代 Windows CE 内核。Windows CE 早期版本是专门面向 PDA 的, 这就是 Windows CE 的第一代内核。到 Windows CE 的第三个版本 (Windows CE 3.0) 才开始支持多种 CPU 架构以及在多个不同领域的应用。其实 Windows CE 3.0 是 Windows CE 第一次重写内核的版本, 这就是 Windows CE 的第二代内核, 跟以前的 Windows CE 1.0、Windows CE 2.0 变化很大, 在任务调度, 实时性方面的支持有了极大地改善。Windows CE 的第四、第五版都是在第二代内核 (Windows CE 3.0 内核) 的基础上产生的。第五版 (Windows CE 5.0) 问世不久, 嵌入式硬件发展相当快, 使用 Windows CE 5.0 开发出现了不少瓶颈, 比如, Windows CE 5.0 最多只支持 32 个进程, 每个进程的访问空间只有 32MB 等, 这些已经满足不了现代的嵌入式产品的需求。因此, 微软对 Windows CE 的第六版 (Windows Embedded CE 6.0) 再次重新编写内核, 在进程和线程控制等方面的架构做了重新设计, 这就是 Windows CE 的第三代内核。据微软对外发布的消息, Windows CE 的第七版 (Windows Embedded Compact) 仍然是基于第三代内核的。

有过驱动程序开发经验的工程师都知道, 为同一个内核的不同版本的操作系统开发设备驱动程序, 其中的开发思路、结构模型都相同, 只是在一些细节上有少量差别。但是, 如果为不同内核的操作系统开发设备驱动, 那差别就很大了, 几乎是开发两套不同的驱动程序。当然有的操作系统在设计的时候就考虑到了移植的问题, 在不同内核版本之间提供了兼容或类似的接口, 相关的移植问题也会变得简单, 比如把 Windows CE 5.0 的驱动或 BSP 移植 Windows Embedded CE 6.0 不是很困难的。

为了帮助读者学习更加成熟并且实用的技术，本书采用基于 Windows CE 的第三代内核来编写。

2.1 本书的编写目的

Windows CE 的设备驱动及 BSP 开发涉及多个领域的知识，它涵盖了微电子技术、电子信息技术、计算机软件和硬件等多项技术。另外，学习 Windows CE 最好具备相应的标准开发板（SDB）和软件（Visual Studio 2005，Platform Builder），还需要有经验的人进行指导开发。目前，国内大部分高校都很难达到这种要求，这也是造成当前国内 Windows CE 开发人才极其缺乏的现象的原因之一。很多希望学习 Windows CE 设备驱动或 BSP 的人已经具备了一定的硬件知识，并且对操作系统原理、数据结构等都很了解，但在 Windows CE 下进行设备驱动或 BSP 开发仍然没有合适的入口点。编写本书的主要目的就是对于那些 Windows CE 设备驱动或 BSP 开发的初学者有个很好的指导作用，让他们少走弯路。

此外，笔者希望通过编写本书来总结这几年来在工作中的研发经验，和更多的读者分享自己的技术，也是对自己所掌握技术的巩固；通过编写本书，笔者更加清楚了实践与理论之间的联系，从而将自己的亲身经验和教训展现在全书的每个章节中。

2.2 本书的主要组成部分

本书由 18 章组成，分三大部分。第一部分是第 1~11 章，主要介绍 Windows CE 下设备驱动程序的开发过程，涉及设备驱动程序开发的基本概念，相关设备驱动实例讲解，设备驱动程序调试、移植等内容；第二部分是第 12~17 章，主要介绍如何开发一个 BSP，涉及 BSP 基本概念，Boot Loader 开发，OAL 开发，相关文件配置等；第三部分是在第一、二部分的基础上进行举例说明，主要介绍一个家庭智能化系统的项目开发。每章的内容概要如下。

（1）第 1 章 Windows CE 驱动简介，对基于 Windows CE 驱动程序的开发做概述。包括驱动的概念，驱动的角色，驱动的分类，什么时候需要开发驱动，如何选择适当的驱动模型来开发自己的设备驱动等内容。

（2）第 2 章 Windows CE 下驱动程序的中断处理，介绍 Windows CE 下的中断处理，中断服务例程（ISR），中断服务线程（IST）的实现方法，并提供相关的示例代码。

（3）第 3 章 Windows CE 下驱动程序的管理，介绍 Windows CE 下的驱动程序如何通过设备管理器来管理，并提供相关的演示代码进行说明。

（4）第 4 章 Windows CE 下驱动程序的访问控制，介绍了驱动开发过程的 DMA 访问，内存访问，总线相关驱动以及总线无关驱动的访问，设备接口等的实现方法。

（5）第 5 章 Windows CE 下流接口驱动程序设计方法，介绍了流接口驱动的工作原理，流接口驱动的实现过程以及流接口函数使用方法的讲解，并演示在 eBox2300SX 开发套件上开发一个标准流驱动的全过程。

（6）第 6 章 Windows CE 驱动程序的调试方法，介绍使用 CETK 测试一个驱动程序的必要软硬件条件，如何使用 CETK 调试一个驱动程序的完整过程，在介绍的过程中分为使用 CETK 提供的模块调试设备驱动程序和使用自定义的模块调试设备驱动程序两种情况，并用相关实例进行演示。

（7）第 7 章 编写 GPIO 驱动程序，先介绍 GPIO 的相关概念、GPIO 的特性、工作原

理,然后以 Vortex86SX 上的 GPIO 为例,详细介绍编写 GPIO 驱动的开发流程,并提供相关的流驱动演示代码,最后介绍 GPIO 驱动实现的另一种方法。

(8) 第 8 章 编写看门狗驱动程序,先介绍传统看门狗的工作原理、操作流程,然后介绍 Vortex86SX 看门狗的一些新特性,以及相应的控制方法,最后详细介绍如何编写看门狗驱动程序。

(9) 第 9 章 编写音频驱动程序,首先介绍音频驱动开发基础知识,包括音频驱动的各种模型及工作原理,然后介绍音频驱动相关注册表的配置、音频驱动程序的移植,最后列举多种音频驱动的开发实例。

(10) 第 10 章 编写块驱动程序,首先介绍块设备驱动开发基础,包括块驱动的架构,块设备管理器,块设备文件系统,块驱动接口,块驱动的加载、安装、侦测、访问等,然后介绍块设备驱动的相关注册表设置,最后介绍 ATADISK 磁盘驱动,ATAPI 驱动的编写方法。

(11) 第 11 章 移植 Windows CE 其他版本的驱动到 Windows CE 6.0,介绍基于 Windows CE 的第二代内核的驱动移植到基于第三代内核驱动的方法以及相关注意事项。

(12) 第 12 章 板级支持包,首先介绍 BSP 的概念,包括 BSP 的由来、BSP 的构成,然后对 Boot Loader、OAL、配置文件、设备驱动程序进行简介,最后介绍创建一个 BSP 的两种方法,从零开始创建一个 BSP 和克隆一个已经存在的 BSP。

(13) 第 13 章 Windows CE 下 Boot Loader 的分析,首先介绍 Boot Loader 应该具备的功能,Boot Loader 开发应该考虑的硬件因素,Windows CE 镜像的引导模式,接着介绍以太网下 Boot Loader 的工作原理与实现方法,并提供相关的代码,最后介绍几种专用 Boot Loader 的制作方法。

(14) 第 14 章 开发 Boot Loader,结合相关的实例代码详细讲解 Boot Loader 开发的全过程。

(15) 第 15 章 Windows CE 下 OAL 的分析,主要介绍 Windows CE 第三代内核的架构,内核的调度,内核的状态,内核独立传输层(KITL)工作原理及其实现过程,可变时钟调度器的实现过程,产品质量级 OAL 等内容。

(16) 第 16 章 如何开发 OAL,结合相关实例详细介绍了 Windows CE 第三代内核 OAL 的设计架构,OAL 启动过程,开发 OAL 的流程。

(17) 第 17 章 创建 Windows CE 配置文件,主要介绍源代码配置文件,包括 Dirs 文件、Makefile 文件、Sources 文件的作用以及构建方法,Windows CE 镜像配置文件[二进制镜像构建文件(.bib)、注册表文件(.reg)、文件系统文件(.dat)、数据库文件(.db)、字符串文件(str)]的构建方法。

(18) 第 18 章 项目实例——家庭智能化系统开发,主要分析如何从零开始开发一个基于 Windows CE 的项目。

致 谢

首先,感谢 DM&P 集团 R&D 总工程师陈志源先生,是他把我领进 Windows CE 的开发领域,并经常对我做技术上的点拨,使我在技术的道路上不断地成长,同时感谢他在百忙之中审阅了本书,并提出了宝贵的修改建议,并为本书作序。其次还要感谢西北工业大学自动化学院副教授张新家老师,感谢他为本书提出了不少建设性的建议,并为本书作序;感谢长

安大学的研究生徐波，西安电子科技大学的研究生孙海浪，解放军理工大学研究生朱利利三位朋友在我写书期间帮我整理相关资料并做了大量的工作。另外，还要感谢我的父母、哥哥、张文英、张秀峰、简清舟、蔡俊娜、葛成敬、贾英、温红光、李双路、周国、周庆、卢一、孔洋、陆荣展、陈克喜、庞永平、王少颖、范贤均、骆吉航、张宏军、管磊、王帮坤、梁万、陈方、高奇、马焱、孟刚、陈蕾、牛广阔、马战国、王引强、宋世光、刘一平、席文涛、杨海刚、王丽娜、苟军、赵建芳、田秀婷等对我的支持，在我写书最困难的时候，是他们鼓励我坚持了下来。还要特别感谢 DM&P 集团的戴文钟、邱英祯等前辈的大力支持，感谢 ICOP 所有同事的支持，最后感谢中国电力出版社的编辑，在从写书到出版的过程中提出了不少有价值的参考意见，让我不断地完善这本书。

限于笔者水平，加之时间仓促，书中疏漏与不妥之处在所难免，希望广大读者不吝指教。欢迎大家登录我的个人论坛来做技术交流：<http://www.embeddedsoft.cn/bbs>

周建设

2009 年 5 月于西安



跋

近年来,我国嵌入式系统应用产品日益丰富,市场呈现快速增长趋势,尤其在家电、电子、汽车、通信、交通、金融、网络、监控、工业自动化、航天航空、军工等领域应用更加广泛,发展嵌入式系统技术和产业已经成为贯彻“信息化带动工业化,工业化促进信息化”方针的强有力手段,它的蓬勃发展使我国的嵌入式产品由“中国制造”向“中国创造”迈出了关键性的一步。

在当今的嵌入式领域中,涉及的 RTOS 种类繁多,大家耳熟能详的有 Windows CE, Embedded Linux、VxWorks、QNX、pSOS 等。这些 RTOS 都各有特色,无法笼统地说谁优谁劣,但是在目前的市场上,大部分的 RTOS 都是微软嵌入式的产品,微软为客户提供的友好、易用的开发环境、统一的版本管理以及一流的技术支持,这些无论是哪种 RTOS 都无法与之相媲美的。Windows CE 从第三代内核起,就开始百分之百地共享内核源码,其内核源码从此向专业的开发人员和学术机构完全免费开放,这将极大地推动 Windows CE 在各行业的应用和发展,促进其不断完善和推广。Windows CE 凭借自己友好的图形界面、完全免费开放的内核源码、统一的版本管理以及一流的技术支持等特点,在嵌入式领域中拥有得天独厚的发展优势。

然而,尽管 Windows CE 在嵌入式领域有着得天独厚的优势,可是在其底层开发却仍然有很大的难度,尤其是在设备驱动及其 BSP 开发方面,对于一个初学者来说是一件很有难度的工作,而市面上相关的参考书籍和资料却寥寥无几,因此一本质量高、实用性强的参考书籍对开发者来说,可以起到事半功倍的效果。本书就是在这种情况下应运而生的。此书是基于 Windows CE 第三代内核的版本,作者周建设将自己在 Windows CE 领域多年的开发经验拿出来与大家分享,以微软嵌入式十大金牌合作伙伴之一、每年微软全球大学生嵌入式竞赛硬件提供商——ICOP(昭营科技)设计的 SoC(Vortex86 SX\DX)系列板卡为硬件平台,深入浅出地介绍了 Windows CE 设备驱动及 BSP 开发的基本原理与开发流程,并提供了大量的实例与开发技巧。这是一本很有参考价值的 Windows CE 开发书籍,可以作为高等院校电子类、电气类、控制类、计算机类专业高年级本科生、研究生学习嵌入式 Windows CE 设备驱动及 BSP 开发的辅助教材,也可供广大嵌入式爱好者、希望转入嵌入式领域的科研和工程技术人员参考使用。

张新家

于西北工业大学



目 录

序

前言

第 1 章	Windows CE 驱动简介	1
1.1	什么是驱动	1
1.1.1	驱动的概念	1
1.1.2	驱动的角色	2
1.2	驱动的种类	2
1.2.1	单层驱动和分层驱动	2
1.2.2	内置驱动和流驱动	4
1.2.3	用户和内核模式驱动	5
1.3	何时需要驱动	6
1.4	驱动开发的步骤	6
1.5	Windows CE 共享的驱动代码	7
第 2 章	Windows CE 下驱动程序的中断处理	8
2.1	Windows CE 中断的相关概念	8
2.2	中断处理	9
2.2.1	中断服务	9
2.2.2	中断屏蔽	10
2.2.3	嵌套中断处理	10
2.2.4	共享中断处理	10
2.2.5	中断处理过程分析	11
2.2.6	配置、注册、并撤消一个中断处理程序	13
2.3	中断服务例程	14
2.3.1	安装 ISR 和设备驱动程序	15
2.3.2	安装一个 ISR 并等待中断事件的产生	16
2.3.3	编译一个 ISR 错误分析	16
2.3.4	安装 ISR 和内核	17
2.4	中断服务线程 (IST)	22
2.5	中断通知	25
第 3 章	Windows CE 下驱动程序的管理	27
3.1	设备管理器的角色	27
3.2	设备管理器的架构	28
3.3	设备管理器相关注册表键	31
3.4	设备文件名称	31
3.5	I/O 资源管理器	33
3.6	设备管理器相关 API 介绍	35

3.6.1	设备管理器枚举类型	35
3.6.2	设备管理器常用函数介绍	36
第 4 章	Windows CE 下驱动程序的访问控制	37
4.1	DMA	37
4.1.1	Windows CE 传统处理方法	37
4.1.2	Windows CE 处理 DMA 的新方法	38
4.2	驱动程序的内存访问	40
4.3	总线无关驱动程序	43
4.4	总线相关驱动程序	48
4.4.1	总线命名	49
4.4.2	总线驱动相关注册表键介绍	49
4.4.3	总线驱动电源回调	50
4.5	总线枚举器	51
4.6	设备接口类	52
4.7	设备接口通知	54
第 5 章	Windows CE 下流接口驱动程序设计方法	57
5.1	流接口驱动程序简介	57
5.2	流接口驱动程序架构	58
5.3	流接口驱动程序的实现	58
5.3.1	流接口驱动程序入口点	58
5.3.2	单通道和多通道	59
5.4	流接口函数介绍	59
5.5	开发一个标准流驱动程序的完整流程	63
5.5.1	创建驱动 STD	63
5.5.2	注册表中注册设备驱动程序	68
5.5.3	将设备驱动程序打包到 Windows CE 的镜像文件中	69
5.5.4	创建设备驱动程序的.def 文件	69
5.5.5	编写驱动测试程序	70
5.6	编写串口驱动程序	72
5.6.1	串口工作原理分析	72
5.6.2	编写串口驱动程序的准备工作	73
5.6.3	填写相关流接口函数	74
5.6.4	小结	75
第 6 章	Windows CE 驱动程序的调试方法	76
6.1	使用 CETK 测试一个驱动程序的必要软硬件条件	76
6.2	使用 CETK 测试一个驱动程序的完整过程	77
6.2.1	使用 CETK 提供的模块测试设备驱动程序	77
6.2.2	使用自定义的模块测试设备驱动程序	86
第 7 章	编写 GPIO 驱动程序	95
7.1	Vortex86SX SoC 简介	95

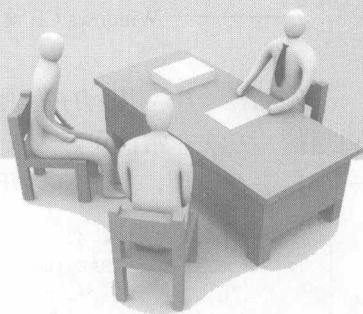
7.2	GPIO 相关概念介绍	95
7.2.1	GPIO 的概念	95
7.2.2	GPIO 的特性	96
7.2.3	Vortex86SX 的 GPIO 简介	96
7.2.4	Vortex86SX 的 GPIO 相关寄存器介绍	97
7.3	GPIO 驱动程序的编写	98
7.3.1	GPIO 工作流程分析	98
7.3.2	编写串口驱动程序的准备工作	98
7.3.3	填写相关流接口函数	98
7.3.4	创建.def 的文件	100
7.3.5	在注册表中注册 GPID 驱动程序	100
7.3.6	编写 GPID 驱动的测试程序	101
7.4	编写完标准 GPIO 驱动程序后的反思	102
7.4.1	GPIO 驱动程序的另一种写法	102
7.4.2	用 C#使用 GPIO (调用 SX_GPIO.DLL)	104
7.5	小结	110
第 8 章	编写看门狗驱动程序	111
8.1	看门狗简介	111
8.1.1	看门狗的概念	111
8.1.2	看门狗的工作原理	112
8.1.3	看门狗的操作流程	112
8.2	eBox2300SX 看门狗介绍	113
8.2.1	Vortex86SX 第一只内置看门狗 WDT0 介绍	113
8.2.2	Vortex86SX 第二只内置看门狗 WDT1 介绍	118
8.3	编写看门狗驱动程序	121
8.3.1	Vortex86SX 内置看门狗工作流程分析	121
8.3.2	创建 Windows CE 驱动工程	122
8.3.3	实现 Vortex86SX 内置看门狗驱动程序	123
8.3.4	创建.def 的文件	125
8.3.5	编写 WDT_DLL 驱动子工程的注册表文件	126
8.3.6	编写 WDT_DLL 驱动子工程的 WDT_DLL.bib 文件	126
8.3.7	编写 WDT_DLL 驱动子工程的 Sources 文件	126
8.3.8	编译 WDT_DLL 驱动程序子工程	127
第 9 章	编写音频驱动程序	128
9.1	音频驱动开发基础	128
9.1.1	音频压缩管理器驱动程序	129
9.1.2	音频驱动的 MDD 层和 PDD 层	131
9.1.3	可插拔的编/解码器的要求	132
9.1.4	统一音频模型	133
9.1.5	Wavedev2 音频驱动模型	135

9.2	音频驱动程序注册表设置	136
9.2.1	ACM 驱动程序的注册表设置	136
9.2.2	软混频器注册表设置	136
9.3	音频驱动移植	137
9.4	音频驱动实例	137
9.4.1	ACM 驱动程序实例	137
9.4.2	Wavedev 驱动程序	153
9.5	音频驱动相关函数、结构体、消息介绍	154
第 10 章	编写块驱动程序	157
10.1	块设备驱动程序开发基础	157
10.1.1	块驱动架构	157
10.1.2	块驱动管理器	157
10.1.3	块设备文件系统	157
10.1.4	块驱动程序接口	158
10.1.5	加载块驱动程序	158
10.1.6	安装块驱动程序	159
10.1.7	侦测块驱动程序	159
10.1.8	访问块驱动程序	159
10.1.9	块设备的电源循环处理	159
10.2	块驱动程序注册表设置	160
10.2.1	自动加载块设备的名称限制	160
10.2.2	PC 卡接口块驱动注册表设置	160
10.2.3	内置块驱动的注册表键设置	161
10.2.4	ATAPI 块驱动注册表键设置	161
10.2.5	SD 卡注册表设置	168
10.2.6	硬盘驱动程序的命名	168
10.3	块驱动程序实例	168
10.3.1	ATADISK 磁盘驱动实例	168
10.3.2	ATAPI 驱动实例	171
第 11 章	移植 Windows CE 其他版本的驱动到 Windows CE 6.0	186
11.1	驱动移植访问权限的确认	187
11.2	驱动移植的 Marshalling	188
11.3	驱动移植的安全复制	191
11.4	驱动移植的线程访问许可	191
11.5	驱动移植的用户接口	192
11.6	驱动程序的移植实例	192
第 12 章	板级支持包	194
12.1	BSP 基本概念	194
12.1.1	BSP 的由来	194
12.1.2	BSP 的构成	194

12.2	Boot Loader	195
12.3	OAL	195
12.4	运行时镜像的配置文件	195
12.5	设备驱动程序	196
12.5.1	SoC 驱动程序	197
12.5.2	BSP 驱动程序	197
12.5.3	通用驱动程序	198
12.6	如何创建一个 BSP	198
12.6.1	从零开始开发个性化 BSP 的流程	199
12.6.2	通过克隆相似平台的 BSP 开发个人 BSP 的流程	207
第 13 章	Windows CE 下 Boot Loader 的分析	211
13.1	Boot Loader 应该具备的功能	211
13.2	Boot Loader 开发时考虑的硬件因素	212
13.3	Windows CE 镜像的引导模式	212
13.3.1	Boot Loader 引导模式	212
13.3.2	非 Boot Loader 引导模式	213
13.4	以太网下的 Boot Loader	214
13.4.1	以太网下 Boot Loader 代码的执行	214
13.4.2	启动和初始化代码的执行	217
13.4.3	构建 Boot Loader 镜像相关文件	218
13.4.4	编译 Boot Loader 镜像	218
13.5	专用的 Boot Loader	219
13.5.1	x86 BIOS Boot Loader	219
13.5.2	x86 串口 Boot Loader	228
13.5.3	x86 ROM Boot Loader	228
13.6	Boot Loader 支持库	228
13.6.1	BLCOMMON 代码库	228
13.6.2	eboot 代码库	229
13.6.3	Bootpart 代码库	229
13.6.4	以太网调试库	229
第 14 章	开发 Boot Loader	231
14.1	Boot Loader 的设计	231
14.2	设立命令行构建环境	232
14.3	为 Boot Loader 的 StartUp()函数创建一个文件	233
14.4	创建 Boot Loader 的 Makefile 文件和 Sources 文件	233
14.5	编译 Boot Loader 源代码	234
14.6	实现 Boot Loader 的 StartUp()函数	234
14.7	创建 Boot Loader 的 Main()函数	236
14.8	为 Boot Loader OEM 函数创建占位程序	236
14.9	编辑 Boot Loader 的 Sources 文件	237

14.10	创建 Boot Loader 的 .bib 文件	237
14.11	重新构建 Boot Loader	239
14.12	实现串口调试函数	239
14.13	创建驱动程序全局缓冲区和启动参数	242
14.14	实现 OEMPlatformInit()函数	244
14.15	实现与以太网控制器相关的函数	251
14.16	实现 OEMPreDownload()函数	253
14.17	为下载 Windows CE 镜像创建一个测试 BIN 文件并验证	256
14.18	实现 OEMLaunch 函数	256
14.19	Flash 相关存储器函数的实现	261
14.19.1	OEMMapMemAddr()函数介绍	261
14.19.2	OEMWriteFlash()函数介绍	262
14.19.3	OEMIsFlashAddr()函数介绍	263
14.19.4	OEMShowProgress()函数介绍	263
14.19.5	Flash 存储器擦除操作相关函数介绍	263
14.20	添加存储器确认支持	264
第 15 章	Windows CE 下 OAL 的分析	266
15.1	内核架构分析	266
15.2	内核调度	267
15.2.1	线程调度计时器	267
15.2.2	内核全局变量调度	267
15.3	内核状态	268
15.3.1	挂起状态	268
15.3.2	OEM 断电状态	269
15.3.3	唤醒状态	269
15.3.4	OEM 唤醒状态	269
15.4	内核独立传输层	270
15.4.1	KITL 传输通信	270
15.4.2	中断和轮询传输	270
15.4.3	KITL 的工作模式	271
15.4.4	以太网调试服务	272
15.5	可变时钟调度器的实现	275
15.6	产品质量级 OAL	277
第 16 章	如何开发 OAL	279
16.1	Windows CE 的 OAL 设计框架	279
16.1.1	OAL 结构 OEMGLOBAL 分析	281
16.1.2	Kernel 结构 NKGLOBAL 分析	286
16.2	OAL 的启动过程分析	290
16.2.1	初始化 CPU	290
16.2.2	内核初始化	292

16.2.3	初始化调试串口	298
16.2.4	初始化硬件平台	298
16.2.5	第一个线程调度前重新初始化内核	307
16.2.6	启动内核	307
16.2.7	第一线程调度后重新初始化内核	310
16.3	开发 OAL 的流程	311
第 17 章	创建 Windows CE 配置文件	342
17.1	源代码配置文件	342
17.1.1	Dirs 文件	342
17.1.2	Makefile 文件	343
17.1.3	Sources 文件	344
17.2	CE 镜像配置文件	350
17.2.1	二进制镜像构建文件	350
17.2.2	注册表文件	362
17.2.3	文件系统文件	366
17.2.4	数据库文件	367
第 18 章	项目实例——家庭智能化系统开发	370
18.1	定制 Windows CE 操作系统	370
18.1.1	新建一个项目工程	373
18.1.2	进入定制操作系统的向导	373
18.1.3	选择 BSP	374
18.1.4	选择操作系统设计的模板	374
18.1.5	选择模板中的相关组件	375
18.1.6	添加 Catalog 组件	376
18.1.7	添加外部驱动程序	378
18.1.8	选择编译模式	379
18.1.9	选择 Windows CE 镜像的默认语言支持	380
18.1.10	选择编译选项	380
18.1.11	添加环境变量	381
18.1.12	关闭调试串口	382
18.1.13	生成 Windows CE 操作系统的镜像	384
18.1.14	下载 Windows CE 镜像到目标设备	385
18.2	家庭智能化应用程序的开发	388
18.2.1	为家庭智能化系统的开发安装 SDK	388
18.2.2	开发家庭智能化系统的应用程序	390
参考文献		401
跋		402



Windows CE 驱动简介

当一个嵌入式产品的项目确定要开发时，我们首先应该选择合适的硬件平台，当硬件平台明确之后，就要考虑选用什么类型的嵌入式操作系统。如果选用 Windows CE 作为操作系统，那么进行上层应用程序的开发将会变得非常简单。不过这需要有一个前提条件：要提供与该硬件平台对应的 BSP 和足够的驱动程序。这是 Windows CE 开发的一个技术难点。那么如何给自己的主板开发合适的 BSP 呢？没有研发能力的公司往往把这外包出去，甚至连硬件都是买市场上已有的。当然，如果使用技术功底深厚的公司的板卡，他们提供的 BSP 质量较高，能够避免应用程序开发过程中的不少麻烦。但是，如果由于某种特殊的需要，要求公司自己开发 BSP，那么就有必要分析 BSP 的开发流程，在 BSP 的开发过程中，驱动开发又是一个重头戏，因此本书将用 11 个章节的篇幅介绍驱动开发的相关原理，开发流程并分析一些驱动程序的实例，之后再介绍 BSP 相关部分的开发。

对于一个应用工程师来说，觉得驱动开发非常神秘，没有任何开发思路。其实驱动开发并不是想象中的那么难，对于从事驱动开发的经验非常丰富的人来说，驱动开发基本上就是参考数据手册（datasheet），把相关的数据放到对应的位置这么简单。不过要想成为一个驱动开发的高手，必要的磨炼是不可缺少的。笔者认为学习驱动开发，应从基础开始，不要搞跳跃式学习，否则写出来的驱动程序虽然可以动作，但是在实际工作过程中往往漏洞百出，这样驱动程序通常是经不起严格测试的。下面先介绍 Windows CE 驱动的情况。



1.1 什么是驱动

1.1.1 驱动的概念

设备驱动是从操作系统中提取物理或虚拟设备的软件，从图 1-1 所示 Windows CE 系统架构图中可以看出驱动程序统一管理相关设备的操作。其中物理设备包括声卡、显卡、网卡、触摸屏、USB 设备等，逻辑设备也有很多，如文件系统。

有了驱动程序之后，应用程序开发工程师就不需要理解底层设备的实现细节，直接调用驱动程序提供的 Win32 API 接口函数就可以进行项目开发，那些设备底层的实现细节对于应用程序开发工程师来说是透明的。“驱动”还有另一个特点是能让其他类似的设备为应用程序提供一个共同的接口，虽然这些设备之间可能存在一些差异。

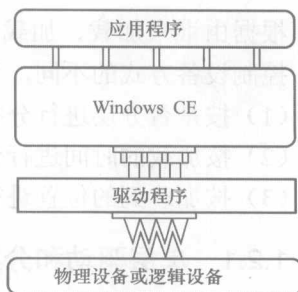


图 1-1 Windows CE 系统架构图