



美国微软出版社授权中文版系列书

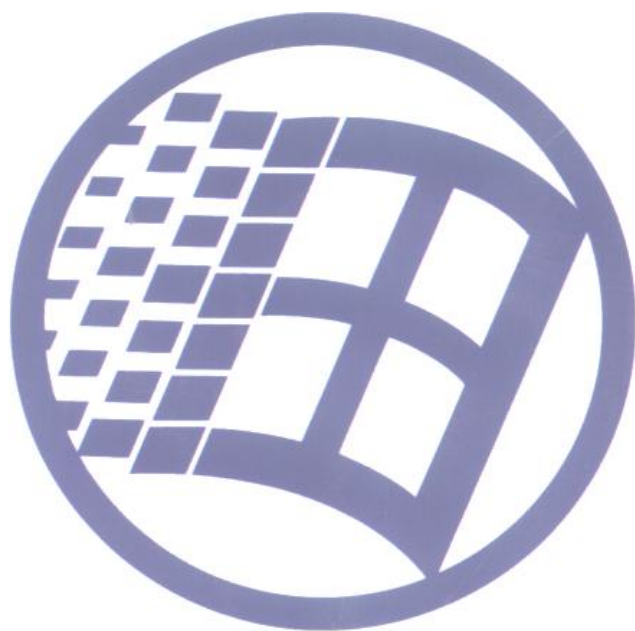
Microsoft Windows CE开发人员指南系列

Windows CE —— 标准、权威、系统、完整的参考手册和开发工具



光盘内容:

1. 本书的英文版电子书
2. SDK for Windows CE,
Handheld PC Professional Edition 3.0
3. 范例源代码
4. 送“精通Visual C++ 5.0”
多媒体学习软件



Microsoft® **Windows® CE** **Device Driver Kit** 设备驱动程序开发指南

〔美〕微软公司 著
希望图书创作室 译



北京希望电脑公司
北京希望电子出版社
www.bhp.com.cn

Microsoft Press



美国微软出版社授权中文版系列书

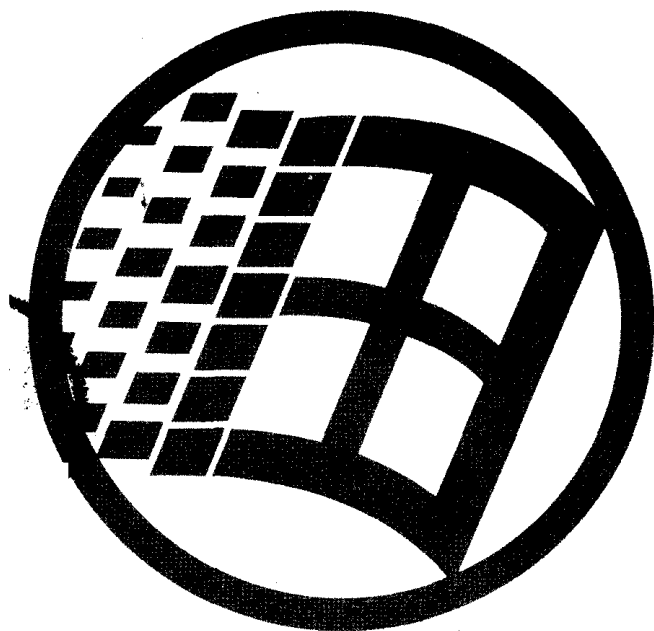
HOPE Microsoft Windows CE开发人员指南系列

Windows CE —— 标准、权威、系统、完整的参考手册和开发工具



光盘内容:

1. 本书的英文版电子书
2. SDK for Windows CE,
Handheld PC Professional Edition 3.0
3. 范例源代码
4. 送“精通Visual C++ 5.0”
多媒体学习软件



Microsoft® **Windows CE** **Device Driver Kit** **设备驱动程序开发指南**

〔美〕微软公司 著
希望图书创作室 译



北京希望电脑公司
北京希望电子出版社
www.bhp.com.cn

Microsoft Press

内 容 简 介

本书是 Microsoft Windows CE 开发人员指南系列书之一,主要讲述创建 Windows CE 的设备驱动程序时所需要的接口、函数结构、消息和其它 API 元素。全书的内容主要包括 Windows CE 设备驱动程序工具包介绍、开发本机设备驱动程序、开发流接口设备驱动程序、声音驱动程序、打印机驱动程序、显示器驱动程序、通用串行总线驱动程序、NDIS 网络驱动程序和块设备驱动程序。

本书按本机设备驱动程序和外设各类设备驱动分类讲解,层次清楚,讲解详细,图文并茂,便于开发人员索引查找。本书适合于计算机系统和设备开发人员、与设备接口开发有关的工程技术人员、高级程序员、Windows CE 爱好者及大专院校相关专业的师生参考使用。

本书配套光盘内容包括:1. 本书的英文版电子书;2. SDK for Windows CE, Handheld PC Professional Edition 3.0;3. 范例源代码;4. 送“精通 Visual C++ 5.0”多媒体学习软件。

版 权 声 明

本书英文版名为“Microsoft Windows CE Device Driver Kit”,由微软出版社出版,版权归微软出版社所有。本书中文版由微软出版社授权出版。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

- 系 列 书: Microsoft Windows CE 开发人员指南系列
书 名: Microsoft Windows CE Device Driver Kit 设备驱动程序开发指南
文 本 著 者: (美)微软公司 著 希望图书创作室 译
责 任 编 辑: 陈河南
CD 制 作 者: 微软出版社 希望多媒体开发中心
CD 测 试 者: 希望多媒体测试部
出版、发 行 者: 北京希望电脑公司 北京希望电子出版社
地 址: 北京海淀路 82 号, 100080
网 址: www.bhp.com.cn E-mail: lwm@hope.com.cn
电 话: 010-62562329,62541992,62637101,62637102 (图书发行,技术支持)
010-62633308,62633309 (多媒体发行,技术支持)
010-62613322-215 (门市) 010-62531267 (编辑部)
经 销: 各地新华书店、软件连锁店
排 版: 希望图书输出中心
CD 生 产 者: 文录激光科技有限公司
文 本 印 刷 者: 北京双青印刷厂
开 本 / 规 格: 787×1092 16 开本 7.625 印张 135 千字
版 次 / 印 次: 1999 年 9 月第 1 版 1999 年 9 月第 1 次印刷
印 数: 0001-5000 册
本 版 号: ISBN 7-900024-56-5/TP·56
定 价: 25.00 元 (1CD, 含配套书)

说明: 凡我社光盘配套图书若有缺页、倒页、脱页、自然破损,本社发行部负责调换

译 者 的 话

本书翻译是希望图书创作室同仁共同努力的结果，全书由袁勤勇负责翻译和统稿，侯志霞、丁岩、沈建军、乔勇、范和平、刘彤及李妍等参与了部分章节的翻译。在此对付出辛勤劳动的各位表示衷心的感谢。

在翻译过程中，我们力求尽善尽美，但错误在所难免，恳请读者指正。



目 录

前言	1	器驱动程序	65
0.1 本书约定	3	6.12 测试显示器驱动程序	66
第一章 Windows CE 设备驱动程序工具包介绍	4	6.13 评测 (profiling) 显示器驱动程序的性能	66
第二章 开发本机设备驱动程序	7	6.14 显示器缓存格式	67
2.1 本机设备驱动程序的系统结构	8	6.15 显示器硬件的推荐	74
2.2 对内部设备驱动程序的限制	9	6.16 显示器驱动程序的注册键	74
2.3 本机设备驱动程序的中断处理	10	第七章 通用串行总线驱动程序	77
2.4 输出样本本机设备驱动程序	13	7.1 USB 拓扑结构	78
第三章 开发流接口设备驱动程序	25	7.2 USB 系统软件	80
3.1 流接口驱动程序的系统结构	26	7.3 编写 USB 设备驱动程序	84
3.2 设备管理程序	27	7.4 样本 USB 设备驱动程序	92
3.3 设备文件名	30	7.5 测试 USB 设备驱动程序	93
3.4 编写流接口驱动程序 DLL	32	第八章 NDIS 网络驱动程序	95
3.5 样本流接口驱动程序	35	8.1 Windows CE 支持的 NDIS	96
第四章 声音驱动程序	37	8.2 编译 Windows CE 的小端口卡驱动程序	97
4.1 样本音频驱动程序	39	8.3 NDIS 小端口卡驱动程序的 DMA 实现	98
4.2 音频压缩管理器驱动程序	40	8.4 NDIS 协议绑定	98
第五章 打印机驱动程序	45	8.5 NDIS 电源管理	99
5.1 打印机驱动程序实现的函数	45	8.6 小端口卡驱动程序的注册表键	99
5.2 打印机驱动程序使用的接口监视器函数	46	8.6 测试 NDIS 小端口卡驱动程序	102
5.3 打印机驱动程序的注册键	46	第 9 章 块设备驱动程序	103
第六章 显示器的驱动程序	49	9.1 用于线性闪存设备的块设备驱动程序	103
6.1 显示器驱动接口	49	9.2 块设备功能	104
6.2 一级显示器驱动程序	50	9.3 线性闪存块设备的局限	105
6.3 二级显示器驱动程序	50	9.4 块设备的系统体系	105
6.4 DDI 函数	52	9.5 块设备文件系统	106
6.5 使用 GPE 类	53	9.6 块设备驱动程序的实现	108
6.6 GDI 支持对显示器驱动程序的服务	53	9.7 原始设备制造商对线性闪存的注意事项	113
6.7 加速位块传送和画线	54	9.8 针对线性闪存的编程注意事项	115
6.8 仿真库中的函数	61	9.9 关于如何在 Windows CE 上使用线性闪存的其它信息	118
6.9 支持抗锯齿字体 (Antialiased Fonts)	64		
6.10 支持 ClearType	64		
6.11 在 OS 映像 (OS Image) 中包含显示			

前言

“Microsoft Windows CE 开发人员指南”系列丛书包含了为基于 Microsoft Windows CE 操作系统的设备开发应用程序所需的全部资料。这一系列丛书包括以下四本书：

- 《Microsoft Windows CE 程序员指南》

介绍了 Windows CE 操作系统的体系结构。

解释了创建基于 Windows CE 的应用程序的底层细节，包括处理进程和线程、管理内存和电源、访问对象库和修改注册表。

介绍了如何把基于 Windows CE 的设备与台式计算机连接，以及如何在设备与台式机之间实现同步化和传输文件。

介绍了如何使用 Unicode 和本地化 Windows CE 应用程序。

- 《Microsoft Windows CE 用户界面服务指南》

介绍了为基于 Windows CE 的设备创建用户界面（UI）的全部相关任务，包括如何创建窗口和对话框，如何处理消息，以及如何增加菜单、控件和其他 UI 资源。

讨论了如何处理不同的用户输入方法（IM），例如键盘和触摸屏。

- 《Microsoft Windows CE 通信指南》

介绍了在基于 Windows CE 的设备上实现通信支持的基本概念，包括如何处理红外连接，如何开发电话应用，如何在应用程序中实现远程访问服务（RAS），如何处理网络互连和安全问题，如何使用 Windows 套接字，以及如何建立 Internet 连接。

- 《Microsoft Windows CE 设备驱动程序开发指南》

介绍了为基于 Windows CE 的设备开发设备驱动程序的过程。

解释了如何创建简单和流接口驱动程序，以及如何实现通用串行总线（USB）和网络驱动程序接口规范（NDIS）驱动程序。

随书附带的光盘包括书的英文电子版，以及如下内容：

内容	描述
Windows CE API 参考手册	说明了接口、函数、结构、消息以及 Windows CE 应用编程接口（API）的其他内容
设备驱动程序包 API	说明了创建 Windows CE 设备驱动程序所需的接口、函数、结构、消息以及其他 API 内容
Windows CE 的 MFC 库	说明了创建完整 Windows CE 应用程序所需的类、全局函数、全局变量和宏

(续表)

内容	描述
Windows CE 主动模板库 (ATL)	说明了开发小巧、快速的 Windows CE 平台 ActiveX 控件所需的类、宏和全局函数
移动频道	描述了如何利用主动服务器页面 (ASP) 和频道定义格式技术实现在基于 Windows CE 的设备上脱机浏览 Web 站点
开发掌上 PC 应用程序	描述了如何使用掌上 PC 的 shell 程序, 如何管理内存和电源, 如何编程访问掌上 PC 导航控件, 以及如何为掌上 PC 应用程序设计用户界面 (UI)
开发手提 PC 应用程序	描述了如何使用手提 PC 的 shell 程序, 如何管理内存和电源, 以及如何在手提 PC 与台式计算机之间同步数据
开发汽车 PC 应用程序	描述了如何实现语音, 如何控制音频系统, 如何与车辆计算机交互, 如何与全球定位系统 (GPS) 通信, 以及如何为汽车 PC 应用程序设计有效的用户界面 (UI)

本书包含下列章节:

Windows CE 设备驱动程序包简介

该章给出了 Windows CE 中使用的各种设备驱动程序模型的概览。

开发本机设备驱动程序

该章提供了有关本机驱动程序的信息, 这些驱动程序具有与 Windows CE 其它组件之间具有特殊目的的接口。

开发流接口设备驱动程序

该章提供了有关为一般设备开发的设备驱动程序的知识。

音频驱动程序

该章提供了 Windows CE 中关于音频压缩管理器驱动程序和波形音频驱动程序的知识。

打印机驱动程序

该章提供了关于 Windows CE 中使用的打印模型和如何为打印机编写设备驱动程序的知识。

显示器驱动程序

该章提供了关于 Windows CE 中使用的显示驱动程序模型和如何为显示设备编写设备驱动程序的知识。

通用串行总线驱动程序

该章提供了下述内容: Windows CE 对 USB 的支持, USB 结构的简要概括, 如何为 USB 设备编写设备驱动程序的知识。

NDIS 网络驱动程序

该章提供了关于 Windows CE 中网络驱动程序接口规范 (NDIS) 驱动程序模型的知识, 以及如何编写网络设备驱动程序的内容。

块设备驱动程序

本章给出了有关 Windows CE 中块设备驱动程序的知识以及如何为块设备，如线性闪存（liner flash memory）设备编写块设备的驱动程序的内容。

0.1 本书约定

下表是本书中使用的字体约定。

约定	说明
斜体	表示占位的最常用的方法和函数参数；表示实现或用户必须提供的信息。例如， <i>lpButton</i> 是一个函数参数。斜体字也表示词汇表中的新术语
大写	表示标志、返回值、消息和属性。例如，WSAEFAULT 是一个 Windows Sockets 错误值，MF_CHECKED 是一个标志，TBADDBUTTONS 是一个消息。此外，大写字母表示用于操作系统命令中的段名、寄存器和术语
()	() 表示语法中用户传递给函数的一个或多个参数

第一章 Windows CE 设备驱动程序工具包介绍

就像其它的操作系统一样，Windows CE 也提供了设备驱动软件，这些软件的目的是驱动内部和外围的硬件设备，或为它们提供接口。设备驱动程序将操作系统和设备链接起来，使得操作系统能够识别设备，并为应用程序提供设备服务。

Windows CE 支持广泛的基于各种 CE 平台的设备驱动程序。也提供一些用于驱动程序开发的模型(model)，其中包括来自其它操作系统的驱动程序模型(model)。因为这些丰富多变的驱动程序模型(model)，Windows CE 适应大部分的内部和外围设备。

Microsoft Windows CE 设备驱动程序工具包配备了文档资料，这些文档资料使得你能够为 Windows CE 创建设备驱动程序。目前，Windows CE 提供了四种设备模型，其中两种是专用于 Windows CE 的模型，另外两种外部模型来自其它操作系统。基于 Windows CE 的两种模型是本机的设备驱动程序和流接口驱动程序。两种外部模型用于通用串行总线(USB)和网络驱动器接口标准(NDIS)的驱动程序。

不同的驱动模型只能通过它们所支持的软件接口来区别，而不是它们所适用的设备。驱动程序模型决定了指定驱动程序输出的软件接口。独立于驱动程序模型，设备驱动程序可以是一个整体也可以是分层的，也就是说设备驱动程序可以直接根据对设备所进行的操作来实现软件接口，也可以将软件接口和针对设备所进行的操作分散到两个层上。由微软提供的许多驱动程序实例都采用了分层结构，因为这样可以减少开发人员在将这些实例移植到别的新设备上时的工作量。如果设备被映射到系统的内存的话，设备驱动程序可以直接访问它们；否则设备驱动程序必须通过下一层的设备驱动程序来访问它们的设备。

比如，PC 卡的设备驱动程序需要用到 PC 卡插槽驱动程序的服务来访问 PC 卡。总之，设备驱动程序可被中断访问；也可以被轮询，或者可能不需要根据设备的状态来更新。

正如它们的名字所表明的，本机设备驱动程序适合于集成到基于 Windows CE 的平台上的设备。设备驱动程序是专为底层内置硬件而设计的，如键盘、屏幕和 PC 卡插槽。因为本机设备驱动程序通常与基于 Windows CE 的平台有着紧密的连接而且每种本机设备驱动程序独有精确的和特殊的目的，微软提供了定制接口的方式来支持内部设备驱动程序。也就是说绝大多数的开发人员不需用编写本机设备驱动程序。但把 Windows CE 定制到新平台的原始设备制造商(OEM)是个例外。他们可以创建他们自己的本机设备驱动程序也可以将微软的本机设备驱动程序的例子移植到他们的基于 Windows CE 的平台上。本机设备驱动程序总是在基于 Windows CE 的平台启动时加载。

与本机设备驱动程序相对应的，具有定制接口和流接口的驱动程序是一般类型的设备驱动程序。流接口驱动程序是在用户一级的动态链接库(DLL)，用来实现一组固定的函数

称为流接口函数，这些流接口函数使得应用程序可以通过文件系统中的特殊文件而与设备进行接口。流接口驱动程序支持几乎任何类型的可以连接到基于 Windows CE 的平台外部设备。比如，开发人员已经设计了流接口驱动程序来支持各种外部设备，其中包括寻呼机、打印机、调制解调器、条码扫描仪和全球定位系统(GPS)接收机。

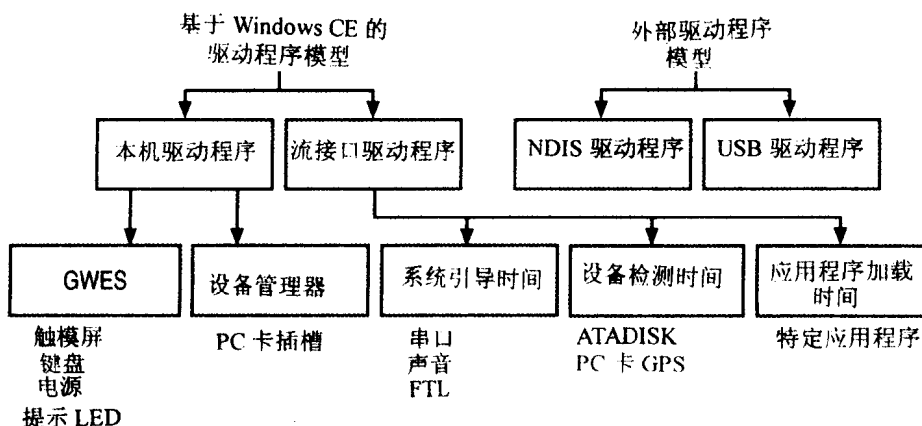
流接口函数可以使应用程序通过文件系统中的特殊文件与设备进行交互。当微软还没有为一些特殊类型的设备定义指定接口时，原始设备制造商(OEM)也可以为这些作为基于 Windows CE 平台一部分的设备创建流接口驱动程序。例如：基于 Windows CE 的销售点(point-of-sale)系统可能在硬件系统中内置了条形码扫描仪，但是因为微软还没有为条形码扫描仪定义定制接口，因此扫描仪的设备驱动程序就是流接口驱动程序，而不是本机驱动程序。

一些普通的内置设备（如串行口）使用流接口驱动程序，这是因为这些设备的功能性更适合流接口驱动程序的结构。为此，原始设备制造商有时写一些流接口驱动程序。为了使他们的外部设备可以在 Windows CE 下使用，外部设备制造商负责提供流接口设备驱动程序。

通用串行总线驱动程序(USB D)将通用串行总线兼容(USB-compliant)设备与 Windows CE 连接起来。与流接口驱动程序不同，通用串行总线驱动程序(USB D)不要求输出任何特定类型的应用函数集。根据所用的设备，通用串行总线驱动程序(USB D)可以输出流接口函数、可以输出定制函数集或使用存在的 Windows CE 应用程序接口来展现设备的功能性。使用已有的 Windows CE 应用程序接口较适合指示设备(pointing device)或大容量存储设备，没有哪种应用程序需要直接使用。Windows CE 有一些现在的机制，输入事件系统和块设备驱动程序应用程序接口分别用于管理这些类型的资源。

网络驱动器接口标准(NDIS)驱动程序采用了 Microsoft Windows NT 操作系统的驱动程序。NDIS 驱动程序是使网络协议生效的驱动程序模型，如 TCP/IP 和红外线数据通信协议，与网卡(NICS)驱动程序的具体实现细节无关。

下图是对当前驱动模型和设备驱动程序的汇总：



本文档的其余部分详细说明了如何实现设备驱动模型。

设备驱动程序工具包包装在两个不同的产品中，即 Microsoft Windows CE Platform Builder 和 Windows CE Platform SDK(软件开发工具包)。其中，Windows CE Platform Builder 工具包括源代码和为 OEM（原始设备制造商）准备的库，各个 OEM 利用这些库建立定制的基于 Windows CE 的平台；Windows CE 平台的 SDK 包括了交叉编译工具和用于建立基于 Windows CE 的平台的必要软件。另外，平台建立工具还包含了源代码和一些样本设备驱动程序。

第二章 开发本机设备驱动程序

要把 Windows CE 移植到目标平台上，必须为在平台上已建立的设备提供驱动程序。一些类型的设备如键盘、显示器和 PC 卡插槽等对操作系统都有一个自定义接口。因为这些接口是专门用于 Windows CE 的，所以这类设备的驱动程序叫做本机设备驱动程序。

一般来说，只是原始设备制造商对本机设备驱动程序感兴趣，他们建立基于 Windows CE 的平台。而独立硬件销售商只开发附加硬件的驱动程序，不需要设计或设置内部设备驱动程序。因此，下述关于内部设备驱动程序的这一部分内容主要是针对原始设备制造商的。

微软为每一类内部设备驱动程序都定义设置接口。尽管每一类内部设备驱动程序各有一个设置接口，然而内部设备驱动程序却为一种特定类型的所有设备提供了一个功能性标准集。这就使得 Windows CE 操作系统可以处理类似的一种特定设备种类的所有情况，尽管有时可能存在硬件的差别。例如，一些基于 Windows CE 的平台把某些种类的 LCD 面板作显示用，但在市场上这些面板有很多类型，在操作特性如分辨率、点位深度 (bit depth)，内存交叉存取(memory interleaving)等方面也各不相同。通过使得所有的显示驱动程序遵循同一种接口，Windows CE 可以用同样的方法处理所有的显示设备，而不管设备自身之间有无物理上的不同。

一般来说，原始设备制造商在建立基于 Windows CE 的平台设置时，都把内部设备驱动程序与图形、视窗和事件子系统 (GWES) 链接起来。但也有例外；在建立过程中并不是所有的内部设备驱动程序都和 GWES 链接。例如，用户安装并加载一些类型的本机设备驱动程序，如显示驱动器和打印机。

但对手持 PC 来说，本机设备驱动程序样本对所有建立在平台上的设备类型都适用；这些设备类型包括：

- 显示器
- 电池
- 键盘
- 触摸屏
- 提示 (notification)发光二极管
- PC 卡插槽

如果目标平台含有的设备集与手持 PC 的设备集不同，就需要为这些设备生成自己的本机设备驱动程序。但如果平台包含的设备与手持 PC 的相似，可以考虑把本机设备驱动程序样本应用到自己的平台上 (Windows CE 平台生成器提供本机设备驱动程序样本)，而不需要再开发自己的本机设备驱动程序。使用这些经过测试的本机驱动程序可以节省时间，并避免错误。即使如此，也没有说必须使用这些样本码来生成内部设备驱动程序或其

它的驱动程序模型，Platform Builder 单独地提供驱动程序样本只是为了方便用户快速开发自己的驱动程序。

下述章节提供的关于本机设备驱动程序的内容用于用户使用和修改平台生成器提供的驱动程序模型。这些章节讨论的内容包括：

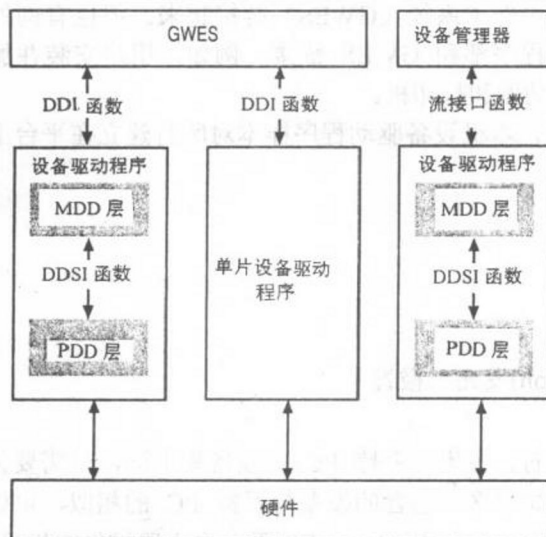
- 本机设备驱动程序的系统结构
- 对本机设备驱动程序和所调用的系统应用的限制
- 本机设备驱动程序中的中断处理
- 本机设备驱动程序样本到用户平台

注 所列目录和路径与 Platform Builder 安装的根目录有关。

2.1 本机设备驱动程序的系统结构

Platform Builder 中包含的样本设备驱动程序分为两种类型：单片(monolithic)驱动程序和分层的(layered)驱动程序。正如其名称表明的，单片驱动程序基于单个码片(single piece of code),该码片直接把硬件设备的功能性传递给操作系统。与单片驱动程序相比，分层的驱动程序由两个设置好的层组成：上层是模型设备驱动程序(MDD)，下层是依赖平台的驱动程序(PDD)。大多数的样本设备驱动程序都配置成分层的，而不是单片的。

下图说明了单片驱动程序和分层的驱动程序是如何在 Windows CE 操作系统中集成的。



微软为链驱动程序提供了模型设备驱动程序(MDD)。对平台和函数来讲模型设备驱动程序(MDD)是通用的，既是原代码也是库。模型设备驱动程序(MDD)执行下列任务：

- 链接 PDD 层并定义它希望调用的函数
- 把不同的函数集提供给操作系统
- 处理像中断处理这样的复杂任务
- 与 GWES 模块和内核通信

每个模型设备驱动程序(MDD)也处理象音频硬件和触摸屏这样的特殊设备集。一般情况下,模型设备驱动程序不需要改变,如果用户宁愿修改 MDD 的话,必须意识到微软并不负责测试、担保或支持用户设置的 MDD。如果微软为修补操作系统的错误或支持最新的版本提供了更新的 MDD,你就必须负责对模型设备驱动程序的后续维护工作。另外,如果你修改了 MDD,你也必须对所有使用了这些 MDD 的 IHV 提供支持。

不象 MDD 层,依赖平台的驱动程序(PDD)层与 MDD 和硬件都有接口,这就意味着必须适合目标平台的需要。一个 PDD 由针对具体硬件的函数组成,而这些函数与一个 MDD 相对应。但这种对应不是直接的一一对应。MDD 函数实现独立的任务,而 MDD 则通过使用这些任务来实现其目标。因为 PDD 是硬件相关的,因此必须生成一个设置好的 PDD 并输出到平台硬件。微软为各种各样的内部设备提供了几个样本 PDD 层。

你也可以不用 MDD 和 PDD 层,把自己的设备驱动程序实现为单片驱动程序。例如,如果性能是一个至关重要的因素,使用单片驱动程序就比使用分层的驱动程序好,因为单片驱动程序避免了与 MDD 和 PDD 层之间的函数调用有关的重叠过程。如果硬件的性能恰好与 MDD 层中的函数所执行的任务相匹配,也可以选择实现单片驱动程序。在这种情况下,实现单片驱动程序可能要比实现分层的驱动程序简单且效率高。但不管你是实现单片驱动程序还是实现分层的驱动程序,都可以基于源代码实现样本分层的驱动程序。

正如上图所示,设备驱动程序接口(DDI)是在 MDD 中实现的函数集,由 GWES 模块调用;设备驱动程序服务提供器接口(DDSI)是在 PDD 中实现的函数集并由 MDD 调用。对单片驱动程序要使用 DDI 函数,而对分层的驱动程序要使用 DDSI 函数。

最后,有些样本设备驱动程序作为流接口驱动程序来执行,也就是说驱动程序把流接口作为它们的 DDI 使用。在这种情况下,不必要把这些驱动程序与 GWES 模块链接起来。它们以普通的 DLL 方式存在并根据需要被调用。音频驱动程序就是一个使用流接口模型的设备驱动程序的例子。因为这些驱动程序具有 MDD 和 PDD,而 MDD 和 PDD 又是开发的基础,因此把它们放到这一节讲述,而没有放到“开发流接口设备驱动程序”一节。

2.2 对内部设备驱动程序的限制

设备驱动程序是用户层次上的过程,也就是说驱动程序代码可以调用 Microsoft Win32 API,访问用户层次可用的任何资源。然而对大多数情况来说,设备驱动程序只需要一定数量的样本 API,比如内存定位程序。驱动程序偶尔也执行比较复杂的任务,包括生成线

程或视窗等。例如在一些基于 Windows CE 的平台上，当电池电压低于临界电压时，电池驱动程序给用户提供一个对话框提示用户更换电池。

驱动程序不能调用 Win32 API 的唯一情况是驱动程序正在处理设备正在关闭的提示。在这种情况下，设备驱动程序不能执行任何能引起其它事件的操作。例如，如果驱动程序现在要打开一个文件，那么核心就需要访问文件系统、调出内存页，并根据不同的执行环境使用其它系统资源等。为了避免这些问题，在关闭系统时限制设备驱动程序的下列动作：

- 在预先定位好的空间存储设备状态
- 向与关闭系统有关的设备发出指令
- 在驱动程序中设置用来提醒其已经关闭的标志

当电源恢复时，驱动程序能从存储数据恢复设备的状态。

驱动程序也能够使用 `SetInterruptEvent` 函数来产生人工中断事件。通过调用 `SetInterruptEvent`，驱动程序可以强行释放中断线程来继续任何必须的事件处理。有关使用 `SetInterruptEvent` 函数的例子请参见 `P2io.c` 文件中的 `HWPowerOn` 函数和 `HWGetIntrType` 函数。驱动程序只应在电源周期变化期间调用这些函数。

2.3 本机设备驱动程序的中断处理

象其它的计算机结构一样，当设备需要驱动程序的服务时，基于 Windows CE 的平台要使用中断通知操作系统。例如在用户敲击键盘或屏幕时，键盘硬件或触摸屏就产生一个中断信号。

然而 Windows CE 不支持嵌套中断，也就是说几个中断不能同时产生。一个中断的处理结束后方可开始另一个中断。尽管这限制了 Windows CE 的实时计划的能力，但却极大地简化了写中断处理器的过程。

Windows CE 把中断处理分为两部分——核心码部分和用户码部分，这样使得功能和实现的简易性比较平衡。核心码部分叫做中断服务例行程序（ISR），而用户码部分叫做中断服务线程（IST）。中断服务例行程序是不可以嵌套的那部分；因此，要使你的中断服务例行程序又短又快，以使得不能使用中断的可能性尽量的小。中断服务例行程序常驻 OEM 适配层（the OEM adaptation layer）并能直接访问硬件注册表。它的唯一任务是确定要把哪个中断标识如 `SYSINTR_SERIAL` 返回给核心中断处理器。

实质上，中断服务例行程序把物理中断映射为逻辑中断。平台无关中断标识符（`Platform_independent interrupt identifier`）声明在头文件 `Nkintr.h` 中，而平台依赖中断标识符（`Platform_dependent interrupt identifier`）声明在平台头文件中，如 `Platform\Cepc\Inc\Oalintr.h`。

由于中断服务例行程序(ISA)较小,并且只做很小的处理工作,因此中断处理器调用中断服务线程(IST)执行大多数的中断处理。中断服务线程(IST)在从函数 `WaitForSingleObject` 得到中断已经发生的信号前一直保持空闲。在中断服务线程(IST)接到信号后,它就在本机设备驱动程序的 PDD 层调用子程序,这些程序反过来访问硬件以获取信息或设置硬件的状态。中断服务线程(IST)是标准第二线程(standard second thread),这些线程可以由操作系统预先清空或调度。为了提高性能,Windows CE 给中断服务线程(IST)使用了两个最高的线程优先级, `THREAD_PRIORITY_HIGHEST` 和 `THREAD_PRIORITY_TIME_CRITICAL`。这些优先级不是专门用于中断服务线程(IST)的,任何第二线程都可以拥有它们。

在基于 Windows CE 的参考平台上,由于使用了 Hitachi SH3 微处理器,系统可以在 2 到 5 微秒内启动一个中断服务例行程序(ISR)。系统在 90 到 170 微秒内启动相应的中断服务线程(IST)。受其它的因素如 CPU 类型、时钟频率、总线速度等的影响,许多实际的基于 Windows CE 的平台的响应时间更短。除硬件的性能外,从一个中断事件到另一个中断事件,中断服务线程(IST)的启动时间还受不可预料因素的影响,这些因素有如处理器缓存的状态、中断服务线程(IST)是否恰巧是正在运行的线程等。

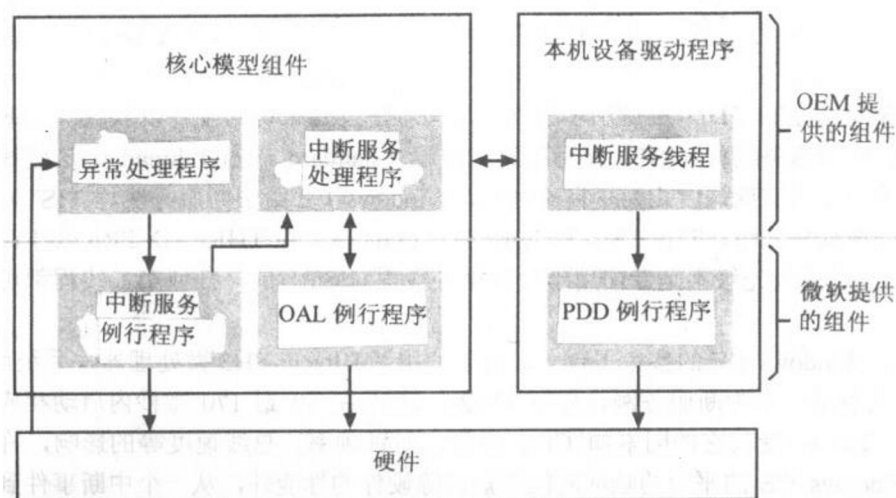
如果你的平台不能满足实时性能的需要,可以考虑下述选择:

- 把更多的处理放入中断服务例行程序(ISR)。但尽量不要这样做,因为中断服务例行程序在运行时,不能安排其它的线程,中断也被屏蔽掉。如果中断服务例行程序运行太长的时间,用户可能体验到很差的设备性能或输入丢失的感觉。
- 增加可以存储几个中断事件数据的硬件并修改中断服务线程,以使得该线程可以把几个中断事件的数据集中到缓冲硬件中。这样就可以使中断服务线程以批处理方式处理中断,从而减少实时需求。
- 让 CPU、数据总线使用高时钟频率或两者都使用。
- 调整平台中断服务线程的优先级。

下图说明了中断处理过程中各部分之间的作用关系。

例外处理程序(exception handler)是所有中断的主要目标。当中断发生时,核心直接跳转到例外处理程序;然后例外处理程序调用已注册的中断服务例行程序处理当前中断。

原始设备制造商(OEMs)在启动时向例外处理程序登记中断服务例行程序。首先,核心在 OAL 中调用函数 `OEMInit`。第二步, `OEMInit` 调用函数 `HookInterrupt` 通知例外处理程序哪个中断服务例行程序对应单独的物理中断线。OAL 中的一些程序如 `OEMInterruptEnable`、`OEMInterruptDisable` 和函数 `OEMInterruptDone` 也在处理过程中调用。



中断处理程序管理向特殊中断注册驱动程序和之后的撤销注册的详细内容。它还维护中断服务例行程序、中断服务线程和 OAL 中的子程序之间的通信。

单片驱动程序也可以使用核心提供的中断事件处理程序支持接口。这个接口是基于标准 Win32 事件的轻量级同步处理模型(lightweight process_synchronization model)。该接口由事件相关的函数如 CreateEvent, SetEvent, ResetEvent, WaitForSingleObject 等和 Windows CE 的核心函数如 InterruptDone, InterruptDisable, InterruptEnable, InterruptInitialize 等组成。分层的驱动程序也用这些同样的函数处理中断, 但你不需要使用这些函数输出(port)分层的驱动程序, 因为模型设备驱动程序(MDD)层不需修改——已经含有对这些函数的适当调用。

下节更详细地讨论如何登记中断处理程序、如何处理以及如何撤销注册等。

2.3.1 注册中断处理程序

在内部设备驱动程序加载后, 驱动程序产生一个中断服务线程并把它注册到中断处理程序中。注册的中断服务线程可用于一个或几个逻辑中断。

如果你为特殊的驱动程序执行一个现成的模型设备驱动程序, 模型设备驱动程序层就为中断注册这个驱动程序。如果你写了一个单片驱动程序, 那么就必须执行一些代码, 从而把驱动程序的中断服务线程注册到中断处理程序中。这些工作可通过使用函数 InterruptInitialize 生成一个事件, 然后把它与中断标识符连接起来实现。

2.3.2 处理中断

在处理中断的过程中会发生一系列具体的事件。你应该为你的内部设备驱动程序写好