



Windows CE Embedded System Development (Based on ARM11)

Windows CE

嵌入式系统程序开发

(基于ARM11)

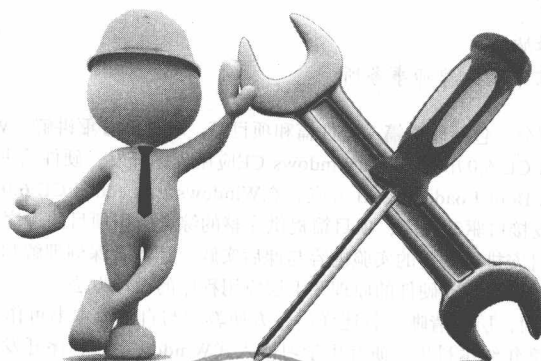
胡文 冯剑 姜海涛 胡玥 编著



机械工业出版社
China Machine Press

TP316.7
490

013032941



Windows CE Embedded System Development (Based on ARM11)



Windows CE

嵌入式系统程序开发

(基于ARM11)

胡文 冯剑 姜海涛 胡玥 编著



北航 C1640737



机械工业出版社
China Machine Press

TP316.7
490

Windows CE嵌入式系统程序开发 (基于ARM11) / 胡文等编著. —北京: 机械工业出版社, 2013.4
(单片机与嵌入式丛书)

ISBN 978-7-111-41915-0

I. W… II. 胡… III. Windows操作系统-程序设计 IV. TP316.7

中国版本图书馆CIP数据核字 (2013) 第057845号

版权所有 · 侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书主要分为3部分, 包括软件篇、硬件篇和项目篇。软件篇着重讲解了Windows CE体系结构、搭建Windows Embedded CE 6.0开发环境和Windows CE应用程序开发; 硬件篇重点讲解了S3C6410硬件体系结构、Windows CE Boot Loader与OAL开发, 在Windows Embedded CE 6.0和Visual Studio 2005开发环境下, 采用C++开发接口驱动程序; 项目篇提供完整的综合应用项目的程序开发实例, 详细介绍了开发过程和源代码。并且安排了丰富的实验内容与课后实践, 让读者深刻理解和掌握嵌入式系统开发的整个过程, 了解底层驱动程序驱动硬件的原理和上层应用程序的设计方法。

本书内容丰富实用, 层次清晰, 叙述详尽, 方便教学与自学。本书可作为高等院校计算机类、电子类和控制类专业高年级本科生、研究生学习嵌入式Windows CE程序开发的教材, 也可作为全国大学生电子设计竞赛培训教材, 以及工程技术人员进行嵌入式系统开发与应用的参考书。

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: 王彬 秦健
北京诚信伟业印刷有限公司印刷

2013年5月第1版第1次印刷

186mm × 240mm · 30印张

标准书号: ISBN 978-7-111-41915-0

定 价: 69.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

前 言

本书以嵌入式操作系统Windows Embedded CE 6.0和ARM11微处理器S3C6410为基础，在软件篇着重讲解了Windows CE体系结构和搭建Windows Embedded CE 6.0应用程序的开发环境，以及使用Windows CE窗体控件开发应用程序。硬件篇介绍了S3C6410的硬件结构，针对相应的硬件目标平台开发Boot Loader和OAL，在Windows Embedded CE 6.0和Visual Studio 2005开发环境下，采用C++开发接口驱动程序。项目篇提供完整的综合应用项目的程序开发实例，介绍了功能模块设计和数据库设计，详细讲解了开发过程和源代码。本书还安排了丰富的实验内容与课后实践，让读者深刻理解和掌握嵌入式系统开发的整个过程，了解底层驱动程序驱动硬件的原理和上层应用程序的设计方法，真正做到了底层驱动的开发与上层应用程序的开发相结合。

本书内容丰富实用、叙述详尽，方便教学与自学。通过介绍结合了DMA-6410XP平台的实验程序，有利于读者掌握嵌入式系统程序开发的方法，并能培养读者综合分析、开发创新和工程设计的能力。通过对本书的学习，读者可以快速提高Windows CE的编程能力和实际开发水平。

全书分三部分，共9章。

第一部分：软件篇

第1章介绍了Windows Embedded CE 6.0体系结构，以及Windows CE系统的内存管理、存储管理与文件系统、进程、线程和中断处理等内容。

第2章介绍了在PC上如何搭建应用程序的开发环境，包括Visual Studio 2005开发工具的安装、Windows Embedded CE 6.0的安装、定制DMA-6410XP系统、执行NK.nb0和Windows CE烧写等。

第3章介绍了本书中第一个Windows CE程序和界面程序设计，使用Windows CE窗体控件创建UI、安装和连接配置SQLCE，以及OLEDB编程访问SQLCE数据库和SQLCE远程数据访问方法。

第二部分：硬件篇

第4章介绍了S3C6410的硬件结构，包括S3C6410微处理器、GPIO输入输出接口、PWM定时器、DMA 控制器、URAT串行接口、IIC总线接口、SPI接口，并举例编写接口电路驱动程序。

第5章介绍了Boot Loader的基本概念、典型结构与启动流程、Boot Loader的实现过程，以及OAL的功能与构成、启动流程。针对相应的硬件目标平台开发Boot Loader，然后介绍了OAL

的开发方法。

第6章在Windows Embedded CE 6.0和Visual Studio 2005开发环境下,采用C++编写接口驱动程序。主要内容包括LED接口及驱动程序、键盘接口及驱动程序、ADC接口及驱动程序、UART接口及驱动程序、GPRS接口及驱动程序、直流电动机接口及驱动程序、步进电动机接口及驱动程序、蓝牙接口及驱动程序等。

第三部分:项目篇

第7章介绍了GPS相关知识、搭建开发环境的方法、采用C++开发GPS定位程序的方法和开发GPS与Google Earth定位程序的方法。

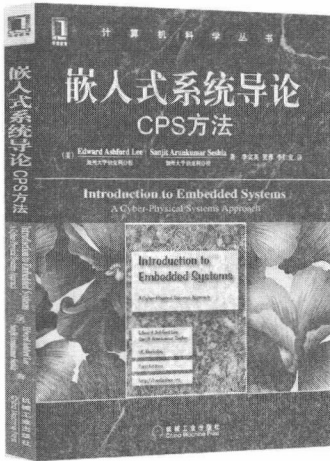
第8章从总体设计着手,介绍了功能模块设计和数据库设计的方法,以及采用C++开发数据传输模块程序和鲜奶配送终端应用程序的方法。

第9章介绍了搭建开发环境和初始化调用环境的方法,以及采用C++开发无线点餐终端应用程序的方法。

本书由胡文教授拟订大纲和目录,并编写了第4章和第6章;冯剑编写了第2章和第3章;姜海涛编写了第1章、第5章和第8章;胡玥编写了第7章、第9章。哈尔滨商业大学的陈铭、李杨、赵艳丽、张凯、陈楠等人为本书的编写做了大量的工作,在此一并表示衷心感谢。本书在编写过程中,参考了大量国内外著作和资料,得到了许多专家和学者的大力支持,听取了多方面的宝贵意见和建议,在此对他们表示衷心感谢。

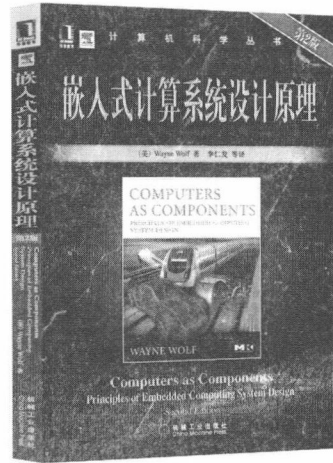
由于时间仓促和水平所限,本书难免有疏漏和不足之处,敬请各位读者批评指正,以期再版时修订。作者邮箱: huwen1957@126.com。

推荐阅读



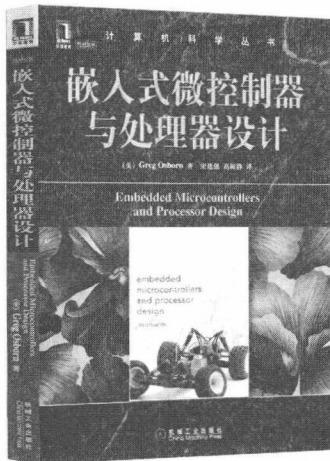
嵌入式系统导论：CPS方法

作者：Edward Ashford Lee 等 ISBN：978-7-111-36021-6 定价：55.00元



嵌入式计算系统设计原理（第2版）

作者：Wayne Wolf ISBN：978-7-111-27068-3 定价：55.00元



嵌入式微控制器与处理器设计

作者：Greg Osborn ISBN：978-7-111-32281-8 定价：59.00元



计算机组成与设计：硬件/软件接口（原书第4版）

作者：David A. Patterson 等 ISBN：978-7-111-35305-8 定价：99.00元

推荐阅读



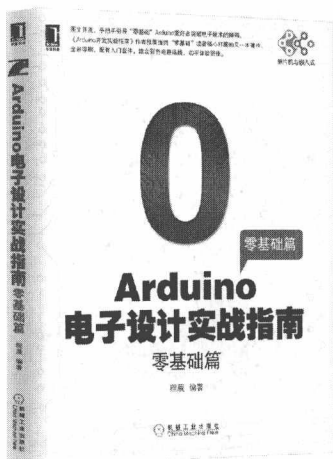
嵌入式系统软硬件协同设计实战指南：基于Xilinx Zynq

作者：陆佳华等 ISBN：978-7-111-41107-9 定价：69.00元



兼容ARM9的软核处理器设计：基于FPGA

作者：李新兵 ISBN：978-7-111-37572-2 定价：69.00元



Ar
作

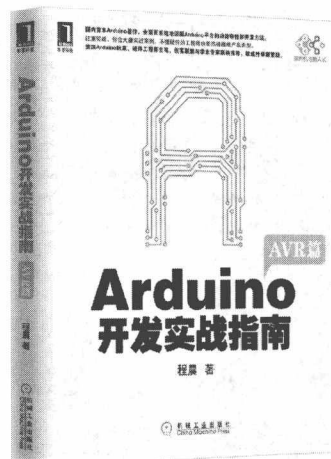


北航

C1640737

：零基础篇

）定价：59.00元



Arduino开发实战指南：AVR篇

作者：程晨 ISBN：978-7-111-37005-5 定价：59.00元

目 录

前言

第一部分 软件篇

第1章 Windows CE系统结构 / 2

- 1.1 Windows Embedded CE 6.0的体系结构 / 2
- 1.2 内存管理 / 7
 - 1.2.1 内存结构 / 7
 - 1.2.2 Windows Embedded CE 6.0 虚拟内存模型 / 8
- 1.3 文件系统与存储管理 / 12
 - 1.3.1 文件系统 / 12
 - 1.3.2 注册表 / 14
 - 1.3.3 存储管理器 / 16
- 1.4 进程与线程 / 19
 - 1.4.1 Windows CE进程 / 19
 - 1.4.2 Windows CE线程 / 21
 - 1.4.3 线程调度 / 24
 - 1.4.4 线程同步 / 26
 - 1.4.5 进程间通信 / 30
- 1.5 中断处理 / 31
 - 1.5.1 中断基本概念 / 31
 - 1.5.2 中断服务例程 / 33
 - 1.5.3 中断服务线程 / 34

第2章 搭建Windows CE开发环境 / 36

- 2.1 安装Visual Studio 2005 / 36

- 2.1.1 安装Visual Studio 2005 / 36
- 2.1.2 安装Visual Studio 2005 Service Pack 1 / 40
- 2.2 安装Windows Embedded CE 6.0 / 42
 - 2.2.1 安装Platform Builder 6.0 Service Pack 1 / 47
 - 2.2.2 安装Windows Embedded CE 6.0 R2 / 50
 - 2.2.3 安装Windows Embedded CE 6.0 R3 / 53
- 2.3 定制DMA-6410 XP系统 / 55
 - 2.3.1 使用项目向导 / 55
 - 2.3.2 构建新的平台 / 57
 - 2.3.3 编辑和修改Windows Embedded CE项目 / 62
 - 2.3.4 添加必要的功能 / 63
 - 2.3.5 Build / 67
- 2.4 执行NK.nb0 / 70
- 2.5 Windows CE烧写 / 73

第3章 Windows CE应用程序开发 / 85

- 3.1 第一个WinCE程序 / 85
 - 3.1.1 SDK的创建与安装 / 85
 - 3.1.2 应用程序设计 / 87
 - 3.1.3 使用C#编写LED驱动程序 / 91
 - 3.1.4 使用MFC编写界面程序 / 93
- 3.2 窗体基本控件 (MFC) / 102
 - 3.2.1 控件的属性、使用方法以及

- 事件 / 103
 - 3.2.2 Edit控件 / 106
 - 3.2.3 CheckBox控件 / 108
 - 3.2.4 RadioButton控件 / 109
 - 3.2.5 ComboBox控件 / 113
 - 3.2.6 ListBox控件 / 119
 - 3.2.7 Spin控件 / 124
 - 3.2.8 Slider控件 / 128
 - 3.3 SQL Server CE / 131
 - 3.3.1 SQLCE简介 / 131
 - 3.3.2 SQLCE 3.0的安装 / 132
 - 3.3.3 SQLCE 3.0连接配置 / 134
 - 3.4 OLEDB编程访问SQLCE 3.0 / 140
 - 3.4.1 OLEDB概述 / 140
 - 3.4.2 OLEDB编程 / 141
 - 3.5 SQLCE远程数据访问 / 151
 - 3.5.1 拉数据 / 152
 - 3.5.2 推数据 / 153
 - 3.5.3 远程T-SQL操作 / 154
 - 3.6 远程数据访问综合示例 / 154
- ## 第二部分 硬件篇
- ### 第4章 S3C6410硬件体系结构 / 170
- 4.1 S3C6410微处理器 / 170
 - 4.1.1 S3C6410结构图 / 170
 - 4.1.2 S3C6410处理器特性 / 170
 - 4.2 GPIO输入输出接口 / 173
 - 4.2.1 GPIO输入输出接口简介 / 173
 - 4.2.2 GPIO专用寄存器 / 174
 - 4.2.3 GPIO接口应用举例 / 183
 - 4.3 PWM定时器 / 185
 - 4.3.1 PWM定时器简介 / 185
 - 4.3.2 PWM的操作 / 185
 - 4.3.3 PWM专用寄存器 / 187
 - 4.3.4 定时器应用举例 / 192
 - 4.4 DMA控制器 / 195
 - 4.4.1 DMA控制器简介 / 195
 - 4.4.2 DMA专用寄存器 / 198
 - 4.4.3 DMA控制器应用举例 / 203
 - 4.5 UART串行接口 / 207
 - 4.5.1 UART串行接口简介 / 207
 - 4.5.2 UART的操作 / 208
 - 4.5.3 UART专用寄存器 / 209
 - 4.5.4 UART接口应用举例 / 215
 - 4.6 I2C总线接口 / 219
 - 4.6.1 I2C总线接口简介 / 219
 - 4.6.2 I2C总线接口操作模式 / 220
 - 4.6.3 I2C总线接口专用寄存器 / 222
 - 4.6.4 I2C总线接口应用举例 / 225
 - 4.7 SPI接口 / 228
 - 4.7.1 SPI总线接口简介 / 228
 - 4.7.2 SPI操作 / 228
 - 4.7.3 SPI专用寄存器 / 229
 - 4.7.4 SPI接口应用举例 / 234
 - 4.8 DMA-6410XP实验平台 / 235
 - 4.8.1 实验平台功能 / 235
 - 4.8.2 外观及接口 / 236
 - 4.8.3 平台硬件配置 / 236
 - 4.8.4 支持操作系统 / 239
- ### 第5章 Windows CE Boot Loader与OAL / 241
- 5.1 Boot Loader介绍 / 241
 - 5.1.1 Boot Loader的引导选项 / 241
 - 5.1.2 Boot Loader的典型特征 / 242

- 5.1.3 Windows CE中Boot Loader为可选项 / 243
 - 5.2 Windows Boot Loader的软件框架和支持库 / 244
 - 5.3 Windows CE操作系统映像文件格式 / 248
 - 5.4 Windows CE的系统启动执行 / 253
 - 5.4.1 OAL的功能 / 253
 - 5.4.2 OAL的构成 / 254
 - 5.4.3 OAL的启动 / 255
 - 5.5 DMA-6410XP中Windows CE OAL开发 / 265
 - 5.5.1 OEMGLOBAL结构体及其整体变量 / 266
 - 5.5.2 OEM地址映像表 / 267
 - 5.5.3 OEMIoControl函数 / 268
 - 5.5.4 OAL的Cache操作OEM函数 / 269
- 第6章 接口驱动程序开发 / 271**
- 6.1 驱动架构分析 / 271
 - 6.1.1 驱动程序分类 / 271
 - 6.1.2 流接口驱动程序架构 / 272
 - 6.1.3 添加一个流驱动到BSP中 / 275
 - 6.2 LED接口及驱动程序 / 276
 - 6.2.1 LED电路设计 / 277
 - 6.2.2 S3C6410 GPIO专用寄存器 / 277
 - 6.2.3 LED驱动程序分析 / 278
 - 6.2.4 LED底层驱动程序设计 / 279
 - 6.2.5 LED驱动程序设计 / 282
 - 6.3 键盘接口及驱动程序 / 293
 - 6.3.1 键盘概述 / 293
 - 6.3.2 矩阵式键盘 / 294
 - 6.3.3 接口专用寄存器 / 296
 - 6.3.4 键盘驱动操作分析 / 298
 - 6.3.5 键盘底层驱动程序设计 / 301
 - 6.3.6 键盘驱动程序设计 / 304
 - 6.4 ADC接口及驱动程序 / 308
 - 6.4.1 S3C6410 ADC / 308
 - 6.4.2 A/D转换接口电路 / 309
 - 6.4.3 ADC专用寄存器 / 311
 - 6.4.4 A/D转换驱动分析 / 315
 - 6.4.5 A/D转换底层驱动程序设计 / 316
 - 6.4.6 A/D转换程序设计 / 320
 - 6.5 UART串口驱动程序 / 323
 - 6.5.1 串行接口概述 / 324
 - 6.5.2 UART接口电路 / 325
 - 6.5.3 UART专用寄存器 / 327
 - 6.5.4 串行通信驱动程序分析 / 331
 - 6.5.5 串行通信底层驱动程序设计 / 332
 - 6.5.6 串口通信程序设计 / 342
 - 6.6 GPRS接口及驱动程序 / 349
 - 6.6.1 GPRS/GSM工作原理 / 349
 - 6.6.2 AT命令 / 350
 - 6.6.3 电话程序设计 / 350
 - 6.6.4 短信程序设计 / 356
 - 6.7 直流电动机接口及驱动程序 / 364
 - 6.7.1 直流电动机驱动电路 / 364
 - 6.7.2 PWM定时器专用寄存器 / 366
 - 6.7.3 直流电动机底层驱动程序设计 / 368
 - 6.7.4 直流电动机驱动程序设计 / 373
 - 6.8 步进电动机接口及驱动程序 / 374

- 6.8.1 步进电动机驱动电路 / 374
- 6.8.2 控制寄存器 / 376
- 6.8.3 步进电动机底层驱动程序
设计 / 377
- 6.8.4 步进电动机驱动程序设计 / 383
- 6.9 蓝牙接口及驱动程序 / 384
 - 6.9.1 蓝牙相关知识 / 384
 - 6.9.2 蓝牙模块的AT命令 / 386
 - 6.9.3 消息 / 388
 - 6.9.4 蓝牙驱动程序设计 / 388

第三部分 项目篇

第7章 GPS与Google Earth定位系统 / 398

- 7.1 GPS相关知识 / 398
 - 7.1.1 卫星星座 / 398
 - 7.1.2 地面监控系统 / 399
 - 7.1.3 信号接收机 / 399
- 7.2 搭建环境 / 400
 - 7.2.1 解析GPS定位信息 / 400
 - 7.2.2 下载并安装Google Earth / 400
- 7.3 GPS定位程序 / 401
- 7.4 GPS与Google Earth定位程序 / 407
 - 7.4.1 Google Earth_Start程序的
建立 / 407
 - 7.4.2 地图定位功能 / 411
 - 7.4.3 在地图上新增地标 / 414

第8章 鲜奶配送系统 / 417

- 8.1 系统总体设计 / 417
- 8.2 搭建开发环境 / 418

- 8.3 系统数据库设计 / 424
- 8.4 系统初始化 / 425
- 8.5 终端应用程序设计 / 426
 - 8.5.1 登录界面 / 426
 - 8.5.2 主界面 / 428
 - 8.5.3 配送模块 / 429
 - 8.5.4 数据传输模块 / 441
 - 8.5.5 历史数据查询 / 444
 - 8.5.6 配送员设置 / 446

第9章 无线点餐系统 / 447

- 9.1 搭建开发环境 / 447
 - 9.1.1 服务器端SQL Server 2005
数据库的安装 / 448
 - 9.1.2 SQLCE数据库的安装与
配置 / 448
 - 9.1.3 SQLCE 3.0连接配置 / 449
 - 9.1.4 OLEDB编程访问
SQLCE3.0 / 453
- 9.2 初始化调用环境 / 460
- 9.3 终端应用程序设计 / 461
 - 9.3.1 系统的登录功能 / 461
 - 9.3.2 系统主菜单 / 462
 - 9.3.3 点餐功能 / 462
 - 9.3.4 结算功能 / 464
 - 9.3.5 查台功能 / 464
 - 9.3.6 转台功能 / 464
 - 9.3.7 更新功能 / 465
- 9.4 系统数据库设计 / 470

参考文献 / 472

第1章 Windows CE系统结构

1.1 Windows Embedded CE 6.0系统结构

Windows Embedded CE 6.0系统结构

第一部分 软件篇

- ◆ 第1章 Windows CE系统结构
- ◆ 第2章 搭建Windows CE开发环境
- ◆ 第3章 Windows CE应用程序开发

Windows Embedded CE 6.0系统结构

Windows Embedded CE 6.0系统结构图如下所示：

Windows Embedded CE 6.0系统结构图展示了系统的层次结构。顶层是Windows Embedded CE 6.0系统，它由多个核心组件组成，包括：

- 操作系统内核（OS Kernel）
- 设备驱动程序（Device Drivers）
- 用户界面（User Interface）
- 应用程序（Applications）

此外，系统还包含以下关键组件：

- Windows Embedded CE 6.0系统库（Windows Embedded CE 6.0 System Libraries）
- Windows Embedded CE 6.0系统服务（Windows Embedded CE 6.0 System Services）
- Windows Embedded CE 6.0系统接口（Windows Embedded CE 6.0 System Interfaces）

这些组件共同构成了Windows Embedded CE 6.0系统的完整架构，为嵌入式设备提供了稳定的运行环境和丰富的功能支持。

第1章 Windows CE系统结构

1.1 Windows Embedded CE 6.0的体系结构

Windows Embedded CE 6.0的体系结构如图1-1所示。



图1-1 Windows Embedded CE 6.0的体系结构

Windows CE 5.0的体系结构是一种分层式结构，从底层向上分别为硬件层、OEM层、操作系统层和应用层。而Windows Embedded CE 6.0划分为用户模式（User Mode）层和内核模式（Kernel Mode）层，其中CoreDLL等DLL同时出现在两个层中，驱动程序也可以加入到内核模式层中，以前的.exe可执行模块基本上都变成了.dll模块。Windows Embedded CE 6.0与之前的Windows CE 5.0体系结构的变化相对较大。

Windows CE 5.0被设计成一种围绕服务而存在的用户模式的程序，称为PSL（Process Server Library，程序服务库），其中NK.exe在内核模式下执行，而操作系统的其他部分则像单独的程序各自执行在用户模式下，例如文件系统Filesys.exe、图形窗口和事件子系统GWES.exe、驱动管理器Device.exe。这样分开的设计让操作系统更加健壮，但为整个操作系统提供主

要功能的服务提供者却以不同程序的身份出现。如果要使用某操作系统提供的服务，则至少会发生一次程序调用，甚至一个简单的操作系统提供的函数调用都不例外。这种设计对系统的效率影响是比较大的。

而Windows Embedded CE 6.0却不同，它将所有操作系统需要提供的服务转移到系统内核的虚拟机（Kernel's Virtual Machine）。这样做的好处是，当系统程序调用服务时，服务程序已经变成了内核的一部分。然而这么做也造成了一些不稳定的机制，例如将驱动程序加载到内核等。Windows Embedded CE 6.0默认在内核模式下执行驱动程序，虽然这样做提高了系统的效率，但如果驱动程序不稳定，将对系统的整体稳定性产生非常严重的影响。当然，并不是所有的驱动程序都是在内核模式下执行的，在Windows Embedded CE 6.0安装完成之后，有一部分驱动程序就是在用户模式下执行的，这样更有利于系统的安全，但是以牺牲设备的性能为代价的。图1-2展示了Windows Embedded CE 6.0里的系统模块。

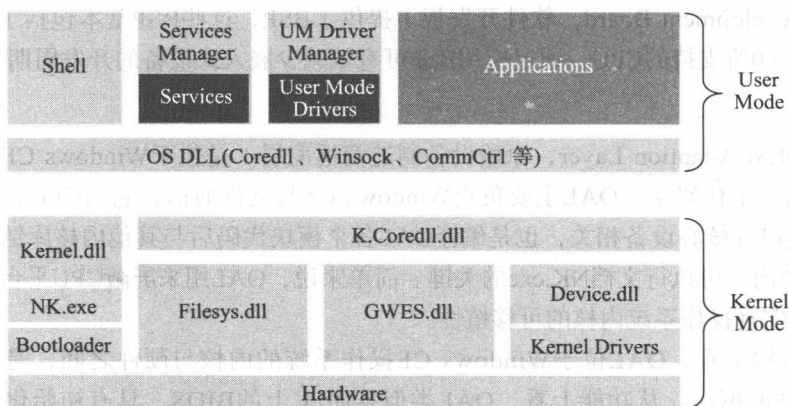


图1-2 Windows Embedded CE 6.0系统模块

如图1-2所示，Windows CE 5.0的各种系统模块，例如Filesys.exe、Device.exe、GWES.exe等，在Windows Embedded CE 6.0中变成了Filesys.dll、Device.dll、GWES.dll，只有NK.exe还是原来的名字。在Windows Embedded CE 6.0中这些服务已经成为系统程序调用（system call）。虽然NK.exe的名字没有变，但其已经区别于Windows CE 5.0中的NK.exe了。Windows CE 5.0中NK.exe提供的各种功能将由Kernel.dll来替代，NK.exe中仅包含一些OAL程序和保证兼容性的程序，这样做的好处是可使OEM和ISV厂商开发的程序与微软提供的Windows Embedded CE 6.0的程序分离，使得内核程序的升级更加容易、更加方便。

从以上叙述中可以看出，Windows CE系统的架构变化是很大的，这对Windows CE的二次开发厂商来说并不是什么好消息，因为这意味着很多建立在旧版本上的驱动，必须进行大规模的重新设计。但微软公司强调这个过程是非常简单的，因为微软公司设计了一个补救措施。这个补救措施是通过图1-2中所示的处在内核模式的K.Coredll.dll来实现的，K.Coredll.dll名字与Coredll.dll非常相似，事实上它就是为了模拟Coredll.dll。K.Coredll.dll仍然在用户模式下执行，

当内核模式下的程序代码调用某些在Windows CE 5.0中处于用户模式下而在Windows Embedded CE 6.0中已转移到内核模式下的API时，K.Coredll.dll会调用这个程序，请求转向内核模式中相应的DLL。K.Coredll.dll并不是唯一有这种机制的DLL，只要其他的DLL需要，也可以采用这种策略。接下来将简要介绍Windows Embedded CE 6.0的系统模块。

1. 硬件

硬件层是指由CPU、存储器、I/O端口、扩展板卡等组成的嵌入式硬件系统，是Windows CE操作系统必不可少的载体。在嵌入式系统领域，因为设备制造商都是由不同领域、不同应用的厂商发展而来的，所以硬件结构相对复杂，仅CPU体系结构就有多种，Windows Embedded CE 6.0支持四种CPU体系结构（ARM、MIPS、x86、SHx）。

硬件的复杂性会增加程序开发的难度，因此出现了BSP（Board Support Package，板级支持包），以此来解决硬件体系结构的差别问题。Platform Builder for CE 6.0为很多常用的SDB（Software Development Board，软件开发板）提供了BSP，这些BSP基本包括了所有Windows Embedded CE 6.0所支持的CPU。通过应用BSP可有效减少嵌入式设备的开发周期。

2. OAL

OAL（OEM Adaption Layer，原始设备制造商适配层）是位于Windows CE内核和目标硬件平台之间的一个代码层。OAL主要负责Windows CE与硬件通信，它与CPU、中断、内存、时钟和调试接口等核心设备相关，也是编译OAL各个模块代码后与其他内核库链接到一起形成Windows CE的内核可执行文档NK.exe的关键。简单来说，OAL用来屏蔽CPU平台的细节和抽象硬件功能，以实现操作系统内核的可移植性。

从逻辑结构上看，OAL位于Windows CE操作系统的内核与硬件之间，是连接Windows CE系统与硬件的枢纽；从功能上看，OAL类似桌面PC上的BIOS，具有初始化设备、引导操作系统以及抽象硬件功能等作用。但与BIOS不同的是，OAL隶属于操作系统，是操作系统的一部分。

3. KITL

KITL（Kernel Independent Transport Layer，内核独立传输层）为我们提供了一种调试Windows CE的简便方法。KITL将通信服务协议和用于通信的硬件分离开，所以我们在创建硬件传输层时就省去很多麻烦，否则我们自己必须实现与设备进行数据交互的协议。KITL工作在硬件传输层之上，因此，它无须关心用于通信的具体硬件，我们可以用USB、串行接口或者以太网接口作为KITL的调试通道，具体选择哪一个，由硬件平台和软件资源决定。有些设备没有以太网接口和串行接口，所以只能采用USB，如Mobile设备。如果系统采用了EBOOT，则建议使用以太网接口作为调试通道。这时，配置KITL的代价相对来说也较小。无论如何，KITL相当强大，在BSP的移植过程中，花一些时间来实现KITL的功能是完全值得的，基于KITL的实现，后续的调试过程中可以节省很多时间。

按启动顺序来说，KITL应该在OAL之后，内核之前启动。所以，必须先完成OAL的移

植，才能进一步移植KITL。

4. 内核

内核为Windows Embedded CE 6.0设备提供最基本的底层功能，这些功能包括服务程序、多线程（Multithread，指从软件或者硬件上实现多个任务并发执行的技术）、内存管理，还包括一些文件管理功能、程序和多线程的调度、实时支持、系统调用、内核电源管理等。NK.exe是OAL进程，内核Kernel.dll是伴随着OAL启动的，由此可知NK.exe是Windows Embedded CE 6.0的核心。图1-3表明了Windows Embedded CE 6.0的内核结构。

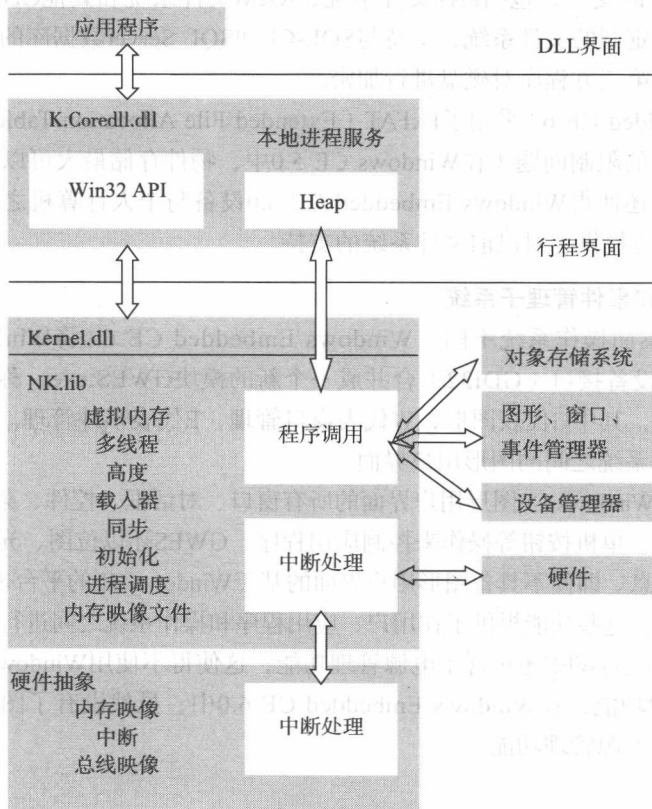


图1-3 Windows Embedded CE 6.0内核结构

图1-3中Kernel.dll的作用类似于Windows CE 5.0中的NK.exe。后面的章节将进一步对Kernel的程序调度、系统调用和存储系统方面进行介绍。

5. CoreDLL

CoreDLL同时出现在用户模式层和内核模式层中，它提供Core OS服务，使得应用程序能够访问Windows Embedded CE 6.0中的资源，如文件系统、内存、设备、程序和线程等。应用程

序也通过这些服务管理和监视其完成任务所需要的资源。通过CoreDLL, 应用程序之间还可以共享程序和其他数据信息。

6. 文件系统

Windows CE文件系统采用了一种灵活的模块化设计, 它允许自定义文件系统、筛选器和多种不同的块设备类型。文件系统和所有与文件相关的API都是通过Filesys.exe进程来管理的, Filesys.exe由ROM文件系统、存储管理器、对象存储三个组件组成。这个模块不但包括文件系统, 还包括对存储对象的管理。Windows Embedded CE 6.0的对象存储系统有丰富的存储功能, 提供对各种文件系统的支持, 包括FAT文件系统、RAM文件系统和其他CD等通用磁盘文件系统。它还支持第三方定制的文件系统, 支持与SQL-CE和SQL Server数据库的连接, 同时拥有良好的安全特性, 支持第三方程序对磁盘进行加密。

Windows Embedded CE 6.0采用了ExFAT (Extended File Allocation Table)。ExFAT不仅解决了大容量文件存储的限制问题(在Windows CE 5.0中, 物件存储最大可以达到256MB, 单个文件不超过4GB), 还使得Windows Embedded CE 6.0设备与个人计算机之间的文件传输更方便、更容易。ExFAT还提供了对以前文件系统的支持。

7. 图形、窗口和事件管理子系统

与Microsoft的桌面操作系统不同, Windows Embedded CE 6.0将Win32 API的用户界面(USER32)和图形设备接口(GDI32)合并成一个新的模块GWES.exe, 称为GWES子系统。GWES是一个缩写词, 其中G代表图形, W代表窗口管理, E代表事件管理。GWES子系统是用户、应用程序和操作系统之间的图形用户界面。

GWES支持组成Windows CE图形用户界面的所有窗口、对话框、控件、菜单和资源, 用户能够通过执行菜单命令、单机按钮等操作来控制应用程序。GWES还以位图、光标、文本以及图标等形式为用户提供信息。即使不具备图形用户界面的基于Windows CE的平台也可以使用GWES的基本窗口和消息功能, 这些功能提供了在用户、应用程序和操作系统之间进行通信的方法。

Windows CE 5.0的GWES还包含了电源管理功能, 这使得不使用Windows CE图形接口的设备无法使用电源管理功能, 在Windows Embedded CE 6.0中, 虽然没有了图形接口的嵌入式设备, 但仍然可以使用电源管理功能。

8. 设备管理器

设备管理器是通过device.dll来实现的。Windows Embedded CE 6.0设备管理器(Device Manager, DM)的职能主要由三个模块完成。内核NK.exe加载device.dll, device.dll加载devmgr.dll, 最后的工作主要由devmgr.dll来完成。DM负责设备驱动的加载、初始化以及卸载。DM将驱动程序的导出接口进行封装, 对应用层提供系统API接口。DM可以在设备发生某特定事件时向应用程序发出通知。DM负责管理设备使用的内存和IO端口等资源。

9. 驱动程序

设备驱动程序是与硬件设备进行通信的系统程序。由于设备驱动程序的存在, 大多数操作