

Windows CE .NET

嵌入式工业用控制器 及自动控制系统设计

叶宏材 陈峙桐 编著

本书主要内容

- 介绍Windows CE.NET应用程序设计基础，藉此说明自动控制程序处理的基本原理，让读者在此基础上自行设计程序。
- 介绍Wincon-8000 Windows CE.NET嵌入式工业用控制器，希望能为读者提供一个进入Wincon-8000更容易的踏脚石。
- 介绍自动控制系统设计及应用实例，希望能为读者提供一个设计自动控制系统的基础平台。



清华大学出版社

Windows CE.NET 嵌入式工业用 控制器及自动控制系统设计

叶宏材 陈峙桐 编著

清华大学出版社

北 京

内 容 简 介

微软 Windows CE 是一个开放且多样化的 32 位嵌入式操作系统。其设计目的是为符合广泛的智能设备的需求,例如从诸如工业控制器、通信集线器和收款机系统(POS)等企业工具到诸如摄影机、电话和家庭娱乐设备等电子消费性产品,为自动控制、视听娱乐、移动计算、终端、数据收集、数据共享及联网等各个应用领域提供一个稳定、实时及多任务的操作系统。Windows CE .NET 被广泛地应用于工业自动化控制器、互联网应用装置、常驻型网关器、个人数字助理等各种设备。

本书希望能提供给读者一个设计自动控制系统的控制平台。此平台可以看作一个简易的自动控制系统架构,全部都设计在一个主程序内,各种功能都以子程序方式设计。由此架构说明自动控制程序处理的基本原理,然后读者由此架构就可以自行设计程序。有此自动控制程序设计的基础后,就可以进一步作套装软件的整合进阶应用。希望本书能提供给读者一个进入 Wincon-8000 更容易的踏脚石。

版权所有,翻印必究。举报电话:010-62782989 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Windows CE.NET 嵌入式工业用控制器及自动控制系统设计/叶宏材,陈峙桐编著.

—北京:清华大学出版社,2005.2

ISBN 7-302-10339-9

I. W… II. ①叶…②陈… III. ①窗口软件, Windows CE—程序设计②自动控制系统—系统设计
IV. ①TP316.7②P273

中国版本图书馆 CIP 数据核字(2005)第 003380 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

责任编辑:闫光龙

封面设计:陈刘源

印 刷 者:北京市清华园胶印厂

装 订 者:三河市金元装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:23 字数:546 千字

版 次:2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷

书 号:ISBN 7-302-10339-9/TP·7038

印 数:1~4000

定 价:30.00 元(含 1 张光盘)

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

微软 Windows CE 是一个开放且多样化的 32 位嵌入式操作系统。其设计目的是为符合广泛的智能设备的需求，例如从诸如工业控制器、通信集线器和收款机系统(POS)等企业工具到诸如摄影机、电话和家庭娱乐设备等电子消费性产品，为自动控制、视听娱乐、移动计算、终端、数据收集、数据共享及联网等各个应用领域提供一个稳定、实时及多任务的操作系统。Windows CE .NET 被广泛地应用于工业自动化控制器、互联网应用装置、常驻型网关器、个人数字助理等各种设备。

本书主要内容包括：Windows CE.NET 应用程序开发；Windows CE.NET 调试环境；基于 Windows 工业用控制器简介；泓格 Wincon-8000 简介；激活及设定 Wincon-8000；开发 Wincon-8000 应用程序；Wincon-8000 应用程序范例；用 eVC++ 4.1 设计自控程序的基础；Wincon-8000 基本 API 功能的应用；自动控制系统基本发展平台。本书将着重于在应用程序开发部分的介绍。

本书希望能提供给读者一个设计自动控制系统的控制平台。此平台可以看作一个简易的自动控制系统架构，全部都设计在一个主程序内，各种功能都以子程序方式设计。由此架构说明自动控制程序处理的基本原理，然后读者由此架构就可以自行设计程序。有此自动控制程序设计的基础后，就可以进一步作套装软件的整合进阶应用。希望本书能提供一个进入 Wincon-8000 更容易的踏脚石。

由于作者水平有限，编写时间仓促，书中难免有不足之处，希望读者和同仁能够及时指出，共同促进技术进步。

编 者
2004 年秋

信息与电子学科百本精品教材工程

《新编计算机类本科规划教材》编委会

主任委员： 刘乃琦（电子科技大学）

副主任委员： 徐建民（河北大学）

周 娅（桂林电子工业学院）

王 越（重庆工学院）

胡先福（电子工业出版社）

委 员： 陈建铎 陈联诚 陈玉明 丁新民 焦占亚 雷景生 骆耀祖

马立权 任满杰 佟伟光 王 力 肖建华 杨 威 叶核亚

朱玉玺 朱战立 张孟玮 冉 哲 李 岩

编辑出版组

主 任： 胡先福

成 员： 张孟玮 冉 哲 李 岩 王 颖

凌 毅 韩同平 李维荣 张 昱

《新编计算机类本科规划教材》参编单位

（按拼音顺序排列）

- | | | |
|------------|------------|--------------|
| ▶ 北方工业大学 | ▶ 贵州师范大学 | ▶ 山西师范大学 |
| ▶ 北京联合大学 | ▶ 哈尔滨工程大学 | ▶ 陕西科技大学 |
| ▶ 长春大学 | ▶ 海南大学 | ▶ 上海第二工业大学 |
| ▶ 长春税务学院 | ▶ 杭州电子工业学院 | ▶ 上海应用技术学院 |
| ▶ 重庆工学院 | ▶ 河北大学 | ▶ 邵阳学院 |
| ▶ 大理学院 | ▶ 合肥学院 | ▶ 沈阳工程学院 |
| ▶ 大连海事大学 | ▶ 湖北工业大学 | ▶ 首都经济贸易大学 |
| ▶ 大连民族学院 | ▶ 湖南工程学院 | ▶ 太原理工大学阳泉学院 |
| ▶ 大连轻工业学院 | ▶ 华北航天工业学院 | ▶ 武汉工业学院 |
| ▶ 电子科技大学 | ▶ 华南农业大学 | ▶ 武汉科技大学 |
| ▶ 佛山大学 | ▶ 江汉大学 | ▶ 五邑大学 |
| ▶ 广东嘉应学院 | ▶ 金陵科技学院 | ▶ 西安石油大学 |
| ▶ 广东韶关学院 | ▶ 南京工程学院 | ▶ 西安邮电学院 |
| ▶ 广西大学 | ▶ 南京师范大学 | ▶ 孝感学院 |
| ▶ 广西工学院 | ▶ 齐齐哈尔大学 | ▶ 烟台大学 |
| ▶ 桂林电子工业学院 | ▶ 青岛科技大学 | |
| ▶ 贵州工业大学 | ▶ 山西农业大学 | |

目 录

第一篇

Windows CE .NET 应用程序设计基础

第 1 章 Windows CE .NET 简介 2

- 1.1 什么是 Windows CE 2
- 1.2 什么是 Windows CE .NET 3
- 1.3 Windows CE .NET 与严格实时系统 4
 - 1.3.1 “实时”的定义 4
 - 1.3.2 实时系统范例 4
 - 1.3.3 实时系统和实时操作系统 5
 - 1.3.4 Windows CE 是一个严格实时操作系统 5

第 2 章 Windows CE .NET 应用程序开发简介 7

- 2.1 开发环境与工具 7
 - 2.1.1 eMbedded Visual C++ 4.0 7
 - 2.1.2 Visual Studio .NET 2003 9
- 2.2 使用 eMbedded Visual C++ 开发应用程序 10
 - 2.2.1 安装 eMbedded Visual C++ 4.0 10
 - 2.2.2 激活 eMbedded Visual C++ 4.0 10
 - 2.2.3 管理项目和工作区 11
 - 2.2.4 建立应用程序 14
 - 2.2.5 指定文件的路径 15
- 2.3 使用 Visual Studio .NET 开发应用程序 16
 - 2.3.1 The Visual Studio.NET 16
 - 2.3.2 .NET Compact Framework 17
 - 2.3.3 Common Language Runtime (CLR) 17
 - 2.3.4 .NET Framework 类库 17

- 2.3.5 .NET Compact Framework 的类 18
- 2.3.6 开启新项目 18
- 2.3.7 Windows CE 装置应用程序与台式计算机应用程序 19
- 2.4 各种开发工具的比较 21
 - 2.4.1 本机码 21
 - 2.4.2 托管码 22
 - 2.4.3 选择适当的开发方式来开发应用程序 22
 - 2.4.4 开发 Windows CE 装置应用程序的注意事项 24

第 3 章 Windows CE .NET 窗口应用程序 26

- 3.1 窗口 26
 - 3.1.1 注册窗口类 26
 - 3.1.2 建立窗口 29
 - 3.1.3 对话框和消息框 30
 - 3.1.4 显示窗口 30
 - 3.1.5 结束窗口 30
- 3.2 窗口应用程序的主程序进入点: WinMain 31
- 3.3 窗口处理程序: WndProc 33
- 3.4 信息处理 35
 - 3.4.1 接收和分配信息 36
 - 3.4.2 WinMain 函数的信息循环 36
 - 3.4.3 传送信息 37
 - 3.4.4 张贴信息 37
 - 3.4.5 定义新的信息 37
- 3.5 使用控制组件 39
 - 3.5.1 取得和设定对话框中控件的显示文本或整数值 39
 - 3.5.2 启动或停用一个控件 39
 - 3.5.3 传送信息给对话框中的控件 40

3.6 本章范例程序	42	4.3.14 列举键及键内的值	124
第 4 章 Windows CE .NET 应用程序		4.4 处理程序及线程	126
开发进阶	58	4.4.1 程序	127
4.1 串行通信应用程序	58	4.4.2 建立程序	127
4.1.1 串行通信应用程序开发接口	58	4.4.3 PROCESS_INFORMATION	
4.1.2 本节范例程序	59	结构成员	129
4.1.3 打开串行通信端口	60	4.4.4 结束程序	129
4.1.4 设定串行通信端口	61	4.4.5 线程	130
4.1.5 设定逾时参数	63	4.4.6 系统排班	130
4.1.6 写入通信端口	65	4.4.7 建立线程	131
4.1.7 读取通信端口	66	4.4.8 结束线程	132
4.1.8 使用通信事件	66	4.4.9 设定线程优先权	133
4.1.9 关闭串行通信端口	68	4.4.10 设定线程的时间量	133
4.1.10 SerialComm.cpp	68	4.4.11 暂停线程	134
4.2 WinSock 通信应用程序	77	4.4.12 继续执行线程	134
4.2.1 [柏克莱版]风格的 socket		4.4.13 线程等待	135
函数	78	4.4.14 线程同步	135
4.2.2 窗口专属延伸函数	80	4.4.15 使用等候函数去等候一个	
4.2.3 本节范例程序	81	同步对象	135
4.2.4 建立 Winsock 通信应用程序	83	4.4.16 本章范例	136
4.2.5 建立 Winsock 服务器端		4.5 本章附录: 系统错误代码对照表	137
应用程序	84	第 5 章 Windows CE .NET 调试环境	170
4.2.6 建立 Winsock 客户端		5.1 在 PC 端安装 Microsoft ActiveSync	170
应用程序	90	5.2 使用 Microsoft ActiveSync 连接	
4.3 文件系统及登录信息	104	PC 和 Windows CE 装置	171
4.3.1 建立和打开文件	106	5.3 安装 Windows CE 装置制造商所	
4.3.2 读取和写入文件	108	提供的 SDK	173
4.3.3 移动文件指针	109	5.4 使用 Windows CE 装置制造商	
4.3.4 取得文件信息	111	所提供的 SDK 建制应用程序	174
4.3.5 文件时间	112	5.5 调校平台管理器	174
4.3.6 文件大小及其他文件信息	113	5.6 开始调试	175
4.3.7 管理文件系统	114		
4.3.8 登录或注册 API	118	第二篇	
4.3.9 建立或打开键	119	泓格科技 Wincon-8000 Windows CE	
4.3.10 读取注册表值	121	.NET 嵌入式工业用控制器	
4.3.11 写入注册表值	122	第 6 章 Windows Based 工业用	
4.3.12 删除键和注册键值	122	控制器简介	178
4.3.13 关闭键	123	6.1 沿革	178

6.2 基于 Windows 的工业用控制器 与 PLC 的比较	179
6.3 结论	179
第 7 章 泓格 ICPDAS Wincon-8000 系列简介	180
7.1 Wincon-8000 = Industrial PC + PLC	181
7.2 硬件规格	181
7.3 Wincon-8000 嵌入式操作系统功能	183
7.4 Wincon-8000 的编程方式	184
7.5 应用实例	186
7.5.1 Wincon-8000 控制系统架构	186
第 8 章 激活及设定 Wincon-8000	192
8.1 一般设定	192
8.1.1 设定机器名称	192
8.1.2 网络设定	192
8.1.3 其他设定	193
8.2 特殊设定	194
8.2.1 存储设定	194
8.2.2 变更显示器分辨率	194
8.2.3 变更默认目录	194
8.2.4 系统信息	195
8.2.5 设定开机自动运行应用 程序	195
8.2.6 设定串行式触摸屏幕	196
8.3 系统更新	197
8.3.1 OS 映像文件更新	197
8.3.2 CF 卡更新	197
8.3.3 Wincon SDK 更新	197
第 9 章 开发 Wincon-8000 应用程序	198
9.1 安装 Wincon SDK 至 PC 开发 环境	198
9.1.1 按部就班地安装 Wincon SDK	198
9.2 开发第一个 eMbedded Visual C++ 应用程序	200
9.2.1 建立一个以窗体为基础的 应用程序	200
9.2.2 新建立窗体应用程序	201
9.2.3 说明范例文件放置的位置	203
9.2.4 设定编译程序的选项	204
9.2.5 设计一应用程序	205
9.3 开发第一个 Visual Basic .NET 应用程序	210
9.3.1 建立 Visual Basic.NET 新项目	210
9.3.2 增加应用程序参考到我们 的应用程序	211
9.3.3 设计应用程序	213
9.3.4 编译应用程序	215
9.3.5 在 Wincon-8000 平台上 执行应用程序	216
9.4 开发第一个 Visual C# .NET 应用程序	216
9.4.1 建立 Visual C#.NET 新项目 ..	216
9.4.2 增加应用程序参考到我们 的应用程序	218
9.4.3 设计应用程序	219
9.4.4 编译应用程序	221
9.4.5 在 Wincon-8000 平台上 执行应用程序	221
9.5 WinconSDK 应用程序开发 接口(API) 链接库及使用手册	222
9.5.1 系统信息函数	223
9.5.2 软件信息函数	223
9.5.3 数字输入/输出函数	224
9.5.4 看门狗定时器函数	226
9.5.5 EEPROM 读写函数	227
9.5.6 模拟输入函数	227
9.5.7 模拟输出函数	229
9.5.8 3-轴编码器(Encoder)函数	230
9.5.9 2-轴步进/伺服 函数	231
9.5.10 计数器/频繁 函数	235
第 10 章 Wincon-8000 应用 程序范例	238
10.1 范例程序说明	238

10.2 应用本机端集中式并列(Parallel)	
高速 I/O 模块	239
10.2.1 DI_DO_Demo for eCV++:	
(数字输入、输出范例).....	240
10.2.2 AI_AO_Demo for eVC++:	
(模拟输入、输出范例).....	240
10.2.3 I-8090 Demo for eVC++:	
(三轴编码器范例).....	241
10.2.4 I-8091 Demo for eVC++: (两	
轴步进电机控制卡范例).....	241
10.2.5 I-8080 Demo for eVC++: ..	242
10.3 应用本机端集中式串行(serial)	
I/O 模块	242
10.4 用 I-7000 系列远程 I/O 模块	244
10.5 应用 I-87K 4/8 系列扩展单元	244
10.6 综合应用范例程序	245

第三篇

自动控制系统设计及应用实例

第 11 章 用 eVC++ 4.1 设计自控

程序的基础	256
11.1 设计准备工作	256
11.1.1 使用 eVC++ 4.1 产生	
新的项目	256
11.1.2 加入 Wincon-8000 的	
SDK 等	259
11.2 Thread 基本设计方式	260
11.2.1 使用 THREAD_INFO	
架构传送必要系数	261
11.2.2 激活 Thread	261
11.2.3 执行 Thread	262
11.2.4 停止 Thread	263
11.3 异步通信端口的的设计技巧	264
11.3.1 Global 变量使用	264
11.3.2 pen Port Dialog 画面	265
11.3.3 RS232C Thread 程序说明	266
11.3.4 RS232C 通信子程序说明	267
11.4 TCP/IP 网络的设计技巧	268
11.4.1 系统初期设定	269

11.4.2 TCP/IP Dialog 画面	269
11.4.3 TCP/IP Server Thread	
程序	270

第 12 章 Wincon-8000 基本 API

功能的运用	272
12.1 实时日期、时间的取入	272
12.2 模块信号处理	273
12.2.1 系统基本 API	273
12.2.2 AI/AO 模块	274
12.2.3 DI/DO 模块	277

第 13 章 自动控制系统基本发展平台 .. 280

13.1 系统基本规格	280
13.1.1 系统功能说明	280
13.1.2 系统架构图	281
13.2 以 Modbus 规格定义 I/O 信号点	281
13.2.1 I/O 信号种类	282
13.2.2 各信号点及系数定义	
架构	282
13.3 数值 Analog 信号处理	285
13.3.1 信号转换步骤图	285
13.3.2 电器信号种类说明	286
13.3.3 模拟转换数字值的解析度	286
13.3.4 AI、AO 信号值与工程值	
的转换	286
13.3.5 次滤波处理	288
13.3.6 内部数值信号应用	289
13.3.7 警报超限检查	289
13.4 状态 Digital 信号处理	290
13.4.1 DI、DO 信号点	290
13.4.2 延伸性状态信号点	290
13.4.3 信号的输入/输出	
反应速度	293
13.5 Modbus 对外通信功能	294
13.5.1 Modbus 的基本概念	294
13.5.2 Modbus 主要通信功能	295
13.5.3 Modbus 通信子程序	295
13.5.4 odbus Slave 通信程序	298

13.5.5	Modbus Master 通信程序.....	304	13.8.2	Demo8000 使用要点.....	327
13.6	I/O 信号通信程序	309	13.9	PC Utility 功能说明.....	328
13.6.1	I/O 模块介绍.....	309	13.9.1	基本规格	328
13.6.2	通信程序设计	310	13.9.2	画面操作说明	329
13.7	控制用 Timer、Counter 的设计	317	13.9.3	程序说明	338
13.7.1	程序设计说明	318	13.10	基本发展平台功能展示	347
13.7.2	使用方式	321	13.10.1	Sample_1.cpp.....	348
13.8	Demo8000 程序说明	322	13.10.2	Sample_2.cpp.....	351
13.8.1	程序说明	323			

第一篇 Windows CE .NET 应用程序 设计基础

- Windows CE .NET 简介
- Windows CE .NET 应用程序开发简介
- Windows CE .NET 窗口应用程序
- Windows CE .NET 应用程序开发进阶
- Windows CE .NET 调试环境

第 1 章 Windows CE .NET 简介

1.1 什么是 Windows CE

微软 Windows CE 是一个开放且多样化的 32 位嵌入式操作系统。其设计目的是满足广泛的智能设备的需求，例如从诸如工业控制器、通信集线器、和收款机系统(POS)等企业工具到诸如摄影机、电话和家庭娱乐设备等电子消费性产品，为自动控制、视听娱乐、移动计算、终端、数据收集、数据共享及联网等各个应用领域提供一个稳定、实时及多任务的操作系统。一个典型的 Windows CE 嵌入式系统常被订制为一个能达到特定目的，一个轻量化及内置的能果断反应系统中断的操作系统。Windows CE .NET 目前支持四大系列的 CPU 架构(ARM、MIPS、SHx、X86)及超过两百种品牌的 CPU，同时因为轻量化及高度的模块化及客制化，Windows CE .NET 被广泛地应用于设计如下各种设备。

- 移动电话/智能型通话装置(Cell Phone/Smartphone Device)
- 顾客装置(Custom Device)
- 数字影像装置(Digital Imaging Device)
- 工业自动化控制器(Industrial Automation Device)
- 互联网应用装置(Internet Appliance)
- 媒体应用装置(Media Appliance)
- 个人数字助理/移动手持装置(PDA/Mobile Handheld Device)
- 常驻型网关器(Residential Gateway)
- 收款机装置(Retail Point-of-Sale Device)
- 机顶盒(Set-Top Box)
- 上网板(Web Pad)
- 窗口化简易终端(Windows Thin Client)

一个 Windows CE .NET 装置的操作系统由许多单独的组件所组成，微软更提供超过 400 种以上的组件让 Windows CE .NET 操作系统开发者组成所需的操作系统平台。表 1.1 依据每种技术作分类来介绍所提供的组件。

本书将着重在应用程序开发部分的介绍。Windows CE 提供给程序开发者标准 Win-32 API, ActiveX 控件, MSMQ (message queuing), COM(Component Object Model)接口, ATL(Active Template Library)和 MFC 链接库(Microsoft Foundation Classes Library)等熟悉的开发环境。ActiveSync 提供台式计算机和嵌入装置之间容易的(网络)连接，无论是通过序列连接，还是红外线串行端口或网络电缆连接。Windows CE 对多媒体(包括 DirectX)、通信(TCP/IP, SNMP, TAPI, 或者更多)还有安全性提供内置的支持。种种的整合应用程序，包括了小型的 IE 浏览器，针对小型的 Outlook 的收信信箱的客户程序，还有小型的容许用户扩展并自定义现行的系统的 Word expose 对象，和扩展应用程序的功能。

表 1.1 Windows CE .NET 的组件

技术分类	内容及说明
应用程序和服务开发支持 (Applications and Services Development)	提供 ActiveSync、COM、MFC、XML 等技术及支持协助开发应用程序及服务
机板支持套件 (Board Support Packages)	标准发展板的所有软件, 包含客制化调适层(OAL : OEM adaptation layer)和装置驱动程序等
通信服务和联网 (Communication Services and Networking)	提供例如 802.11 networking、Web servers、UPNP(Universal Plug and Play)、TAPI(Telephony API)等技术, 来与其他装置通信或交换数据
核心操作系统服务 (Core OS Services)	Windows CE 操作系统平台的一般性功能和提供系统核心的信息
国际化 (Internationalization)	支持装置本土化(localization), 例如中文的显示及输入
多媒体技术 (Multimedia Technologies)	提供 waveform audio、Microsoft Direct3D、Microsoft Windows Media 和 DVD-Video API 等技术以协助开发快速的影音应用程序
对象存储及登录 (Object Store and Registry)	提供文件系统、对象存储(Object Store)、登录(Registry)系统等技术以协助在平台上储存数据
登录设置 (Registry Settings)	可在登录(Registry)中记录硬件及软件信息
安全服务 (Security Services)	提供例如 CryptoAPI、SSL(Secure Sockets Layer)和 smart card 等技术支持, 有助于应用程序及数据的安全性和保密性
介壳程序及用户接口 (Shell and User Interface)	提供 GDI(graphics device interface)、GWES(Graphics, Windowing, and Events Subsystem)和输入方式等技术, 做为用户、应用程序和操作系统三方间的接口
用户应用程序 (User Applications)	提供标准应用程序, 例如 Microsoft Internet Explorer、Windows Media player、Pocket Office 和游戏等

1.2 什么是 Windows CE .NET

Windows CE .NET 是 Windows CE 3.0 的下一个版本, 从 2002 年推出以后, 目前已有 Windows CE .NET 4.0, Windows CE .NET 4.1, Windows CE .NET 4.2 及 Windows CE .NET 5.0 等版本。而其“.NET”的意思是微软将其近年来在桌面型(desktop)操作系统上

主推的.NET 架构技术(.NET Framework)也在 Windows CE 的嵌入式操作系统上提供,而在 Windows CE 上所推出的.NET 技术又称为“.NET Compact Framework”,顾名思义,是一个精简型的.NET Framework,提供给系统资源有限的嵌入式系统使用。.NET Compact Framework 体现了 Windows CE 装置的.NET Framework 技术,它提供了例如个人数字助理(PDAs)、口袋型装置(Pocket PC)、移动电话(mobile phones)、机顶盒(set-top boxes)、自动控制装置和客制化嵌入装置等资源有限(resource-constrained)的 Windows CE 装置一个非关硬件(hardware-independent)的运行环境来执行各种应用程序。.NET Compact Framework 是.NET Framework 类库(class library)的子集合,它是继承了.NET Framework 的架构重新设计,提供包括通用语言运行时(common language runtime)和托管码运行(managed code execution)、内存管理和资源回收(garbage collection)等功能。.NET Compact Framework 同时提供了例如 Visual Basic .NET/Visual C# .NET 等另一种开发工具语言,并提供 XML 网页服务(XML Web Services)、Microsoft ADO .NET 数据存取、窗口窗体(Windows Forms)和网络通信等应用程序开发功能。

1.3 Windows CE .NET 与严格实时系统

在需要实时反应(time-critical responses)的一些高效率(high-performance)嵌入式应用程序中,实时效率(Real-time performance)常是最重要的要求。例如在工业控制中常需要快速的 I/O、机器人(robotics)及机械等控制,其他领域例如电子通信切换、医学工程仪器、航空和导航仪器等需要实时反应的装置都需要一个能提供开发实时应用程序的软硬件平台。

1.3.1 “实时”的定义

“实时”(real time)分二种情形,即严格实时(或称“硬实时”, hard real time)和轻松实时(或称“软实时”, soft real time)。“实时”的定义有很多种,微软嵌入式操作系统采用 OMAC(Open, Modular, Architecture Control)的定义如下:硬实时系统是一个如果没有达到它的计时需求就会失败的系统,而软实时系统则是当它反应一些类似中断、定时器和排班等系统服务时能容忍较大计时延迟的系统。其英文全文如下: A hard real-time system is a system that would fail if its timing requirements were not met; a soft real-time system can tolerate large latencies in the delivery of operating system services like interrupts, timers, and scheduling.

1.3.2 实时系统范例

我们可以从图 1.1 所示的箱子传送带处理系统的例子来说明“硬实时”和“软实时”的差别。这个系统使用传送带运送箱子,当箱子传感器(box sensor)侦测到输送带送来一个箱子时,会发出中断(interrupt)通知控制器(controller)是否这个箱子是瑕疵品。如果送来的箱子是瑕疵品,控制器将会改变在分流处(divert area)的传送带轨道将瑕疵品往左送。所以,箱子通过箱子传感器一直到分流处之前的这段时间内,如果控制器不能完成判断箱子是否是瑕疵品和做出必要的改变传送带轨道的操作,系统将会因错过对瑕疵品的筛检

而出错。

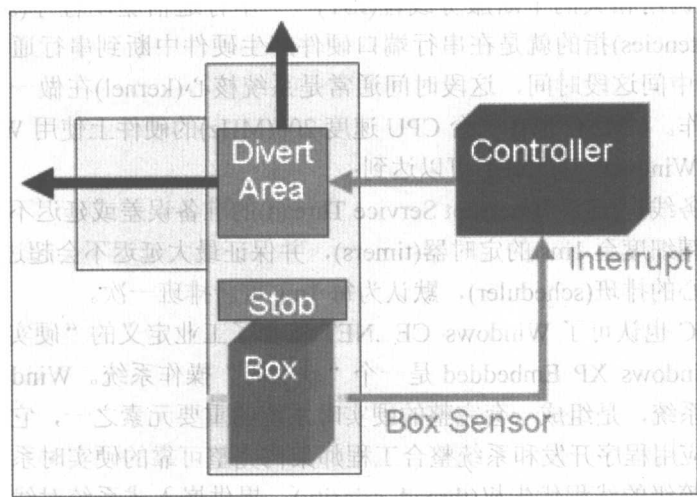


图 1.1 箱子传送带处理系统

介绍完这个箱子传送带处理系统的工作原理后，我们来看看“硬实时”系统和“软实时”系统如何实做这个箱子传送带处理系统。如果控制器是“软实时”系统，在箱子传感器检测到输送带送来一个箱子而发出中断通知控制器的同时，也必须激活停止点(stop)的开关来暂时停止箱子传送带。等到控制器收到箱子传感器送来的中断并完成判断箱子是否是瑕疵品和做出必要的改变传送带轨道的操作后，再由控制器去关闭停止点的开关继续箱子的输送。如果控制器是“硬实时”系统，因为“硬实时”系统能够保证特定的操作能在特定的时间内完成，也就是控制器的处理操作必须在箱子通过箱子传感器一直到分流处之前的这段时间内完成，所以不需要像“软实时”系统一样还需要额外的停止点开关装置和开关操作，不会让传送带停停走走地降低了处理效率。

1.3.3 实时系统和实时操作系统

在此，我们必须区分实时系统和实时操作系统(RTOS: real-time operating system)的差别。实时系统是由所有必要元素所组成，包括硬件、操作系统和应用程序等。而 RTOS 只是组成实时系统的所有必要元素其中之一，必须确保能够称职地协助整套系统达到实时的需求。

1.3.4 Windows CE 是一个严格实时操作系统

OMAC(Open, Modular, Architecture Control)同时研究实验了上百种的应用系统，并发现 95%的应用系统需要 1ms 或比 1 ms 更长的循环时间(cycle times)处理一个事件，并容许有循环时间 10%的“准备误差”(variation)或“准备延迟”(latencies)，即 100μs 或比 100μs 更长。“准备误差”(variation)或“准备延迟”(latencies)指的是，当一个事件发生一直到事件开始被处理的中间这段时间。以 RS-232 串行端口(serial port)通信处理为例子来说明，当有数据送进 RS-232 串行端口时，串行端口硬件会产生硬件中断以通知 Windows

CE .NET 的系统核心(kernel), 然后系统核心会判断中断的种类及做一些初始操作和准备操作, 最后核心会调用相关的中断服务线程(IST)——串行通信驱动程序(serial driver)。而“准备延迟”(latencies)指的就是在串行端口硬件产生硬件中断到串行通信中断服务线程(IST)开始处理的中间这段时间, 这段时间通常是系统核心(kernel)在做一些必要的判断及初始化的准备操作。OMAC 也在一台 CPU 速度 200(MHz)的硬件上使用 Windows CE 来做测试, 结果发现 Windows CE .NET 可以达到:

- 中断服务线程(IST: Interrupt Service Thread)的准备误差或延迟不会超过 100 μ s。
- 可提供精细度至 1ms 的定时器(timers), 并保证最大延迟不会超过 100 μ s。
- 系统核心的排班(scheduler), 默认为每 1ms 重新排班一次。

因此, OMAC 也认可了 Windows CE .NET 符合了工业定义的“硬实时”的需求。而值得一提的是 Windows XP Embedded 是一个“软实时”操作系统。Windows CE .NET 是一个硬实时操作系统, 是组成一套完整的硬实时系统的重要元素之一, 它提供了以下的功能和支持来帮助应用程序开发和系统整合工程师架构完整可靠的硬实时系统:

- 256 个等级的线程优先权(thread priority): 提供嵌入式系统对线程排班的控制能更弹性化。
- 套叠的中断(Nested interrupts): 能先暂停优先权低的中断服务, 让优先权高的中断服务先执行。
- 每个线程的执行时间量(Per-thread quantum): 应用程序可以设置线程排班时所分配到的执行时间量(quantum)。
- 优先权改变或倒置(Priority inversion): 当系统资源例如 mutex、关键区(critical section)和信号(semaphore)等被优先权低的线程取得占用时, 如果同时有其他的优先权高的线程也在等待取得同样的资源才能继续执行, 这将会造成某种程度的竞争现象(racing condition)而严重影响系统整体的效率。因为优先权低的线程必须执行完毕后才释放这些资源, 而此时的执行权是由优先权高的线程所取得, 但是优先权高的线程却必需取得这些资源才能继续执行, 这就造成了互不相让而使系统迟滞甚至于死机。为了解决这个问题, Windows CE 提供了优先权改变或倒置的功能, 当系统核心发现这种现象时, 会提高原本优先权低的线程的优先权, 让它高于原本优先权高的线程, 以取得执行权后赶快执行完并释放资源, 好让其他也需要这些数据的线程接着执行。

在自动化控制领域常需要实时系统, 而在工业领域对实时的定义和需求又更严格。本章先对 Windows CE .NET 做初步的介绍后, 马上介绍实时系统, 和强调 Windows CE .NET 对实时系统支持的功能。本书旨在介绍如何开发 Windows CE .NET 装置或控制器的工业用自动控制应用程序, 因此必须先对 Windows CE .NET 和实时系统做一概述, Windows CE .NET 的核心技术、操作系统(OS: Operating System)的原理和实时系统理论等, 并不是本书所要讨论的重点, 从下一章开始, 讨论重点将会是在 Windows CE .NET 中应用程序的开发与实做。