



ARM嵌入式设计与Windows CE 开发系列丛书

Windows CE 应用程序设计

Windows CE Application Program Design

◆ 张勇 曾炽祥 许波 编著
陈滨 主审

ARM 嵌入式设计与 Windows CE 开发系列丛书

Windows CE 应用程序设计

张勇 曾炽祥 许波 编著

陈滨 主审

西安电子科技大学出版社

2008

内 容 简 介

本书基于目前最流行易用的嵌入式操作系统 Windows CE 平台,以 Windows API 程序框架技术为支撑,以完整的程序实例为导向,详细地阐述了编写 Windows CE 应用程序的方法和技巧。全书共八章,系统地介绍了 Windows CE 嵌入式操作系统下的 Windows API 程序设计框架技术,深入地讲解了 Windows CE 设备的绘图操作、设置字体与子窗口编程、菜单与触摸屏事件的处理、与运算相关的数据处理方法、通用控件与对话框编程技术、常用的文件管理与数据库操作以及 Windows CE 设备辅助输入面板等内容。

本书把握住了 Windows CE 应用程序依靠事件消息驱动的工作机制,结合完整程序实例对窗口过程及消息处理进行了实质性的论述,这也是本书的一大特色。

本书可作为嵌入式开发相关专业或网络与软件工程相关专业的高年级本科生或研究生教材,也可作为从事嵌入式开发的工程技术人员和 Windows CE 程序设计爱好者的参考书。

图书在版编目(CIP)数据

Windows CE 应用程序设计 / 张勇, 曾炽祥, 许波编著. —西安: 西安电子科技大学出版社, 2008.9

ISBN 978-7-5606-2110-4

I. W... II. ①张... ②曾... ③许... 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 119007 号

策 划 李惠萍

责任编辑 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2008 年 9 月第 1 版 2008 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 26.5

字 数 627 千字

印 数 1~4000 册

定 价 40.00 元

ISBN 978-7-5606-2110-4/TP·1082

XDUP 2402001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

Windows CE 操作系统是美国 Microsoft 公司研发的组件化的实时操作系统，目前的 Windows Embedded CE 6 核心部分大小为 300 KB，包括约 700 个组件，支持 ARM、MIPS、SH4 和 X86 等四种嵌入式微处理器，已被广泛应用于掌上 GPS、智能设备、数码相机、便携式媒体播放器、工业控制及遥测等嵌入式设备。右下图为工作在 Windows Mobile 6(即 Windows Embedded CE 5)下的智能手机。

一般地，基于 Windows CE 的嵌入式开发过程可包括以下相对独立又紧密相关的几个方面：

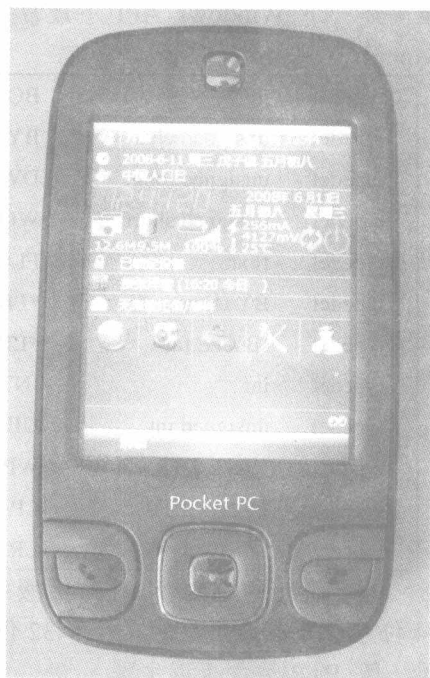
首先，是通用嵌入式硬件平台的设计，即选取 Windows CE 支持的芯片，例如，ARM 核心的 S3C2442，扩展必要的外围电路，例如，64 MB 的 SDRAM、32 MB 的 Flash 和 320 × 240 像素的触摸屏等，构成一块可以加载和运行 Windows CE 操作系统的硬件板卡。当然，除了加载 Windows CE 操作系统之外，还可以针对 ARM 芯片进行芯片级的 C 语言程序设计，此时的 ARM 芯片相当于一片 DSP 芯片。

然后，是 Windows CE 系统的内核定制与加载以及板级的驱动程序开发，即借助 Platform Builder 软件选择开发板支持包(BSP)、选择系统配置结构和设备及服务，生成 Windows CE 操作系统映像和新平台的 SDK；基于 Windows CE 的两种驱动模型，即本机设备驱动和流接口设备驱动，开发实现对特定设备访问的接口软件。

最后，基于设备的 SDK 使用 Embedded Visual C++ 4 SP4 或 Visual Studio 2008 编写设备的应用程序，即编制运行在 Windows CE 操作系统上能处理数据或文件并实现用户要求的特定功能，且具有良好的人机接口界面的程序。这部分内容是本书的重点。

为了系统地介绍基于 Windows CE 的嵌入式开发，我们编写了这套 ARM 嵌入式设计与 Windows CE 开发系列丛书。该套丛书共分四册，分别为《Windows CE 应用程序设计》(即本书)、《ARM 原理与 C 程序设计》、《Windows CE 系统设计》和《Windows CE .Net Framework 程序设计》，将陆续由西安电子科技大学出版社出版。

Windows CE 应用程序设计分为 Windows API 方法、MFC 类库方法和 .Net Framework 方法，出于编程习惯的考虑，我们崇尚 Windows API 方法和 .Net Framework 方法。本书采



Windows Mobile 智能手机

用 Windows API 的程序设计方法进行讲解，该方法的程序设计类似于“搭积木”游戏，首先需要有一个总体性的功能设计；然后，在总的功能设计指导下，搭建第一块“积木”，即创建主窗口；接着，编写第一块“积木”的事件消息处理程序，使“积木”上可以堆积其他“积木”，即在原有的“积木”上设计各种完成特定设计功能的“积木”，这些新的“积木”表现为子窗口和对话框；随后，再为新堆积上的“积木”编写事件消息处理程序，使这些“积木”能响应其事件消息；最后，重复堆积“积木”和编写“积木”的消息处理过程，直到实现所设计的功能。尽管这种“搭积木”式的程序设计编写的应用程序的代码相当长，但是，Windows API 程序设计是公认的编写 Windows CE 程序最有效的方式，目前仍然是其主流的程序设计方法。

Windows API 程序设计隶属于 C 或 C++ 程序设计的范畴，因此，可以将其理解为：

Windows API 程序设计 = 数据结构 + 窗口过程 + 数值算法

数值算法与普通的 C 或 C++ 语言下的运算相同，窗口过程将在第一章中详细介绍，这里阐述一下 Windows API 下最常见的几种数据类型，这些预定义类型包含在 windef.h 和 winnt.h 等头文件中，列举如下：

typedef	int	BOOL;	typedef	wchar_t	WCHAR;
typedef	unsigned char	BYTE;	typedef	WCHAR	TCHAR;
typedef	unsigned long	DWORD;	#define	VOID	void
typedef	unsigned short	WORD;	#define	LPWSTR	LPTSTR
typedef	float	FLOAT;	#define	LPCWSTR	LPCTSTR
typedef	BYTE near	*PBYTE;	typedef	WCHAR	*LPWSTR;
typedef	BYTE far	*LPBYTE;	typedef	HANDLE	HWND;
typedef	int	INT;	typedef	HANDLE	HFONT;
typedef	unsigned int	UINT;	typedef	HANDLE	HPEN;
typedef	UINT_PTR	WPARAM;	typedef	HANDLE	
typedef	LONG_PTR	LPARAM;		HINSTANCE;	
typedef	LONG_PTR	LRESULT;	#define	CONST	const

需要说明的是，带_PTR 后缀的类型，例如，UINT_PTR、LONG_PTR 等，是为了与 64 位系统兼容而定义的，对于 32 位的 Windows CE 系统而言，UINT_PTR 与 UINT 含义相同，即_PTR 后缀可以去掉。

本书内容共分为八章：第一章“Windows CE 程序框架”详细介绍了 Windows CE 应用程序的工作方式，并编写了采用 Windows API 设计 Windows CE 应用程序的框架程序，给出了调试 Windows CE 应用程序的工作环境和编程风格。第二章“绘图、字体与子窗口”和第三章“资源、菜单和触笔事件”讲述 Windows CE 的绘图、字体、子窗口、资源、菜单和触笔事件等内容，通过课表软件实例具体地说明了如何在主窗口和子窗口上绘图、输出字体以及加入菜单操作和触笔事件等内容。第四章“数据处理”集中反映了如何在 Windows CE 环境下进行浮点数的运算，从该章可以看出，Windows CE 系统的数据处理能力毫不逊色，与桌面 Windows 系统相当。第五章“控件和对话框”是 Windows CE 界面设计的基本元素，通过进一步完善课表软件，本章全面讲述了 Windows CE 下界面设计的技巧和方法。第六章“文件管理”和第七章“数据库编程”对 Windows CE 系统下的文件管

理和数据库编程进行了深入的分析，给出的程序实例具有很强的代表性，可以用作记事本软件和学生成绩管理软件。第八章“输入面板”讨论了辅助输入面板的工作方式以及在应用程序中如何使用输入面板等问题。

本书的全部源程序可以通过 Email: zhnyong@21cn.com 向作者索要，但是，我们强烈建议读者自行输入代码并进行调试以提高学习效率。

由于我们的水平有限加之编写时间紧张，尽管我们反复地审阅了每一句话和每一段代码，书中可能仍然会有一些失误，敬请读者来信批评指正。

作 者

2008年5月于江财枫林园

致 谢

经过一年多的时间，本书终于收笔了。本书能够顺利完成并与读者见面，除了封面上的作者们的努力之外，与来自各界的朋友们的帮助是分不开的。

首先需要指出的是，深圳技创科技有限公司(techor.com)伸出了大力援助之手，给我们提供了 Windows CE 实验平台，本书的程序最终都在该平台上测试通过。当然，学习本书，有一台硬件设备或 PPC 智能手机，将会明显增加您学习的乐趣。

我们全体作者一致感谢我们的学校和学院给我们提供了稳定舒适的工作环境。我们软件无线电研究中心的全体老师们为本书的实验做了大量的工作，事实上，他们分担了中心的很多工作，兢兢业业，使得我们可以安心 Windows CE 的开发与研究。他们是陈滨、蔡鹏、黄坚、方勤、邓梅根、谢建宏、彭志宏、邓海、黄国强、朱华贵、冯伟、曾日波、肖晓萍、徐晓兰、罗翠兰、李喜军、黄珍华、汤文芳老师，我们对他们所做的工作表达真挚的谢意。

我们要感谢我们求学时的导师，他们是电子科技大学陈天麒教授和洪时中研究员，导师们对我们学术上和为人处事的言传身教，以及他们做学问和指导我们的身影，时刻指引和鼓励着我们向更高的科技进步。我们感谢与我们生活在一起的爸爸妈妈，他们分担了管理家务和照看孩子的工作，他们任劳任怨，默默无闻地为我们奉献了他们的一切，我们把本书奉献给他们，以表达对他们的尊重和感恩之情。同时，我们感谢我们的其他亲人和兄弟姐妹，她们的亲切鼓励和电话联系，让我们心里充满了信心和温暖。

最后，我们要特别感谢西安电子科技大学出版社的李惠萍编辑，李编辑为我们的每一本书的出版都付出了辛勤的劳动，并具体地做出了指导。

作者

2008年5月

目 录

第一章 Windows CE 程序框架	1
1.1 程序设计环境	1
1.2 Windows CE 程序工作方式	1
1.3 Windows CE 程序框架	1
1.4 程序框架的解释	8
1.5 匈牙利表示法	25
1.6 Visual Studio 开发环境.....	25
1.7 Embedded Visual C++开发环境.....	38
1.8 小结	43
第二章 绘图、字体与子窗口	44
2.1 绘图	44
2.2 课表软件实例一	47
2.2.1 头文件 myTimeTable1.h.....	48
2.2.2 主程序入口文件 myTimeTable1.cpp	50
2.2.3 主窗口过程文件 mainWndProc.cpp	51
2.3 实例一程序解释	58
2.4 课表软件实例二	61
2.4.1 头文件 myTimeTable2.h.....	63
2.4.2 主程序入口文件 myTimeTable2.cpp	64
2.4.3 主窗口过程文件 mainWndProc.cpp	65
2.4.4 子窗口过程文件 tableChildWndProc.cpp.....	70
2.5 实例二程序解释	77
2.6 使用帮助	84
2.7 小结	86
第三章 资源、菜单和触笔事件	87
3.1 资源	87
3.2 触笔事件	94
3.3 课表软件实例三	95
3.3.1 资源文件 myTimeTable3.rc.....	96
3.3.2 头文件 myTimeTable3.h.....	97

3.3.3	主程序入口文件 myTimeTable3.cpp	100
3.3.4	主窗口过程文件 mainWndProc.cpp	101
3.3.5	子窗口过程文件 tableChildWndProc.cpp.....	108
3.4	实例三程序解释	120
3.5	小结	129
第四章	数据处理	131
4.1	控件	131
4.2	计算三角函数实例程序	134
4.2.1	头文件 myTrigonometry.h	135
4.2.2	主程序入口文件 myTrigonometry.cpp	138
4.2.3	主窗口过程文件 mainWndProc.cpp	139
4.2.4	子窗口过程文件 sinChildWndProc.cpp.....	144
4.3	三角函数程序解释	150
4.4	小结	158
第五章	控件和对话框	159
5.1	滚动条控件	159
5.2	滚动条实例	161
5.2.1	资源文件 myTimeTable4.rc.....	162
5.2.2	头文件 myTimeTable4.h	163
5.2.3	主程序入口文件 myTimeTable4.cpp	166
5.2.4	主窗口过程文件 mainWndProc.cpp	167
5.2.5	子窗口过程文件 tableChildWndProc.cpp.....	175
5.3	课表实例四程序解释	190
5.4	对话框与属性表	191
5.4.1	对话框资源	191
5.4.2	模式对话框	193
5.4.3	无模式对话框	195
5.4.4	属性表	196
5.5	对话框实例	202
5.5.1	资源文件 myTimeTable5.rc.....	206
5.5.2	头文件 myTimeTable5.h	209
5.5.3	主程序入口文件 myTimeTable5.cpp	216
5.5.4	主窗口过程文件 mainWndProc.cpp	217
5.5.5	子窗口过程文件 tableChildWndProc.cpp.....	225
5.5.6	“关于”对话框过程文件 myAboutDlg.cpp	242
5.5.7	“课目”对话框过程文件 myCourseDlg.cpp.....	243
5.5.8	“教室”对话框过程文件 myClassroomDlg.cpp.....	247

5.5.9	“教学计划”对话框过程文件 myTeachplanDlg.cpp.....	250
5.5.10	“教学大纲”对话框过程文件 myOutlineDlg.cpp	253
5.5.11	“实验大纲”对话框过程文件 myLibplanDlg.cpp	255
5.5.12	“课程教案”对话框过程文件 myContentDlg.cpp.....	258
5.5.13	“课程讲义”对话框过程文件 myBookDlg.cpp.....	261
5.6	课表实例五程序解释	264
5.7	小结	270
第六章 文件管理		271
6.1	文件读写	271
6.2	文件复制与移动	273
6.3	命令栏图标按钮	273
6.4	Unicode 与 GB 编码	274
6.5	文件管理实例	275
6.5.1	资源文件 myFileManagement.rc.....	280
6.5.2	头文件 myFileManagement.h.....	281
6.5.3	主程序入口文件 myFileManagement.cpp	286
6.5.4	主窗口过程文件 mainWndProc.cpp	287
6.5.5	子窗口过程文件 fileChildWndProc.cpp	293
6.5.6	“关于”对话框过程文件 myAboutDlg.cpp	303
6.5.7	“复制文件”对话框文件 myCopyToDlg.cpp	304
6.5.8	“移动文件”对话框文件 myMoveToDlg.cpp.....	309
6.6	文件管理实例解释	313
6.7	小结	318
第七章 数据库编程		319
7.1	数据库	319
7.2	数据表	320
7.3	查看列表控件	326
7.4	数据库实例	329
7.4.1	资源文件 myCensusInfo.rc.....	331
7.4.2	头文件 myCensusInfo.h.....	334
7.4.3	主程序入口文件 myCensusInfo.cpp	340
7.4.4	主窗口过程文件 mainWndProc.cpp	341
7.4.5	子窗口过程文件 dbChildWndProc.cpp	347
7.4.6	“关于”对话框过程文件 myAboutDlg.cpp	368
7.4.7	“添加或编辑记录”对话框过程文件 myEditDlg.cpp.....	369
7.4.8	“查找记录”对话框过程文件 myFindDlg.cpp.....	376
7.5	数据库实例解释	378

7.6 小结	380
第八章 输入面板	381
8.1 键盘	381
8.2 辅助输入面板	382
8.3 输入法实例	382
8.3.1 资源文件 mySIP.rc.....	384
8.3.2 头文件 mySIP.h.....	385
8.3.3 主程序入口文件 mySIP.cpp.....	389
8.3.4 主窗口过程文件 mainWndProc.cpp	390
8.3.5 子窗口过程文件 sipChildWndProc.cpp.....	398
8.3.6 “关于”对话框过程文件 myAboutDlg.cpp	409
8.4 输入法实例解释	409
8.5 小结	411

第一章 Windows CE 程序框架

进入 Windows CE 程序设计领域之前，要求读者对基本的 C 语言程序设计有所了解，至少对 C 语言的语法和数据结构有所了解。Windows CE 是微软公司开发的专用于嵌入式设备的小体积图形用户界面操作系统，相对于桌面 Windows XP 或 Vista 而言，由于其体积小，支持的 API(应用程序接口)函数相对较少，因此，在程序设计方法上有诸多不同之处。

1.1 程序设计环境

Windows CE 程序设计平台是基于 Visual Studio 2005/2008 或 Embedded Visual C++ 4.0 SP4 开发环境的，同时，需要具有 Windows CE 操作系统模拟器或智能设备模拟器，特别是 Pocket PC 2005 模拟环境或 Windows Mobile 6 模拟环境。当然，具有一个嵌入式开发平台或一部 PPC 智能手机进行最终的程序测试也是必需的。

1.2 Windows CE 程序工作方式

Windows CE 程序采用所谓的消息响应工作方式，这也是桌面 Windows 程序的工作方式，即 Windows CE 操作系统管理设备的一切资源。当有事件发生时，例如，触摸笔点击屏幕等，这些通称为事件，Windows CE 操作系统会产生相应的消息，并把这些消息发送到相应的窗口；窗口在收到这些消息后，再通过一种所谓的“回调”(Callback)方式，指示 Windows CE 操作系统调用相应的消息处理过程，进而完成对该事件消息的处理。

这种所谓的“回调”是指窗口收到 Windows CE 发送来的消息后并不直接处理该消息，而是简单地向 Windows CE 操作系统进行一次确认工作，Windows CE 确认后直接调用相应的消息处理过程或函数。

在本书中，我们把普通 C 程序中的函数和过程通称为“过程”，后文出现的“窗口过程”是特指窗口的消息处理回调过程。同时，我们把过程的定义称为过程的函数原型或过程原型，而过程体或函数体是指过程的具体实现代码。

1.3 Windows CE 程序框架

笔者得益于 Douglas Boling 先生的《Programming Microsoft Windows CE .NET》一书，

当然，10年前从Petzold先生的书中也得到了类似的启发。Windows CE程序设计具有明显的框架特性，任何复杂的程序设计都可以以此框架为基础，通过添加窗口和消息处理过程来完成。

下面，先把这个框架程序源码罗列出来，这个程序框架主要应归功于Douglas先生，笔者做了一些工作，以使这个程序更适合于中国的学生。稍后，在1.4节将详细地解释这些程序代码。

头文件 myFirst.h

```
// fileName: myFirst.h
// by:      zhnyong@21cn.com

// 所有 Windows CE API 程序必须包括的头文件
#include <windows.h>

// 窗口顶部命令栏所需要的头文件
#include <commctrl.h>

// Go: 可以在此处加入你所需要的其他头文件

// 返回数组 x 的元素数
#define dim(x) (sizeof(x)/sizeof(x[0]))

// 定义消息 ID 号与消息处理过程的结构体
typedef struct{
    UINT uCode;
    LRESULT (*functionName)(HWND, UINT, WPARAM, LPARAM);
} MessageProc;

// 定义菜单或控件的命令 ID 号与命令处理过程的结构体。本程序中没有用到
typedef struct{
    UINT uCode;
    LRESULT (*functionName)(HWND, HWND, WORD, WORD);
} CommandProc;

// 定义一个全局正整数标号，这里用于表示窗口顶部的命令栏控件 ID 号
#define IDC_CMDBAR 1

// 注册主窗口类
int registerWnd(HINSTANCE);

// 创建主窗口并显示
int showMainWnd(HINSTANCE, LPWSTR, int);

// 退出主窗口，即退出程序
int quitMainWnd(HINSTANCE, int);
```

```
// 主窗口过程的原型，窗口过程必须是回调函数
LRESULT CALLBACK mainWndProc(HWND, UINT, WPARAM, LPARAM);

// 主窗口收到 WM_CREATE 消息后的处理过程原型
LRESULT goCreateMain(HWND, UINT, WPARAM, LPARAM);

// 主窗口收到 WM_PAINT 消息后的处理过程原型
LRESULT goPaintMain(HWND, UINT, WPARAM, LPARAM);

// 主窗口收到 WM_DESTROY 消息后的处理过程原型
LRESULT goDestroyMain(HWND, UINT, WPARAM, LPARAM);

// 主窗口收到 WM_CLOSE 消息后的处理过程原型
LRESULT goCloseMain(HWND, UINT, WPARAM, LPARAM);
```

主程序入口文件 myFirst.c

```
// fileName: myFirst.c
// by: zhnyong@21cn.com

// 该文件为主程序入口文件，即程序执行的入口点，格式固定
// 包括的头文件，其中，有 windows.h 头文件和其他所有需要的头文件
#include "myFirst.h"

// 主程序过程以 WinMain 为入口
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
                  LPWSTR lpCmdLine, int nShowCmd)
{
    // 定义一个消息结构体对象变量
    MSG msg;

    // 判断注册主窗口成功否，不成功，返回值 1 并退出
    if(!registerWnd(hInstance))
        return 1;

    // 判断创建主窗口并显示主窗口成功否，不成功，返回值 2 并退出
    if(!showMainWnd(hInstance, lpCmdLine, nShowCmd))
        return 2;

    // 获取 Windows CE 操作系统发来的消息，并翻译和回发
    while(GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}
```

```
    }  
  
    // 退出主窗口  
    return quitMainWnd(hInstance, msg.wParam);  
}
```

主窗口过程文件 mainWndProc.c

```
// fileName: mainWndProc.c  
// by: zhnyong@21cn.com  
  
// 该文件为主窗口的各种消息处理子过程，是实现程序功能自定义的过程  
  
// 包括自定义头文件  
#include "myFirst.h"  
  
// 定义全局常量字符串类名  
const TCHAR mainClassName[] = TEXT("myFirst");  
  
// 定义全局程序实例句柄  
HINSTANCE hInst;  
  
// 定义全局常量消息 ID 号与消息处理过程指针查找表  
const MessageProc mainMessages[]={  
    WM_CREATE, goCreateMain,  
    WM_PAINT, goPaintMain,  
    WM_DESTROY, goDestroyMain,  
    WM_CLOSE, goCloseMain,  
};  
  
// 注册主窗口类  
int registerWnd(HINSTANCE hInstance)  
{  
    // 定义窗口类结构体对象  
    WNDCLASS wc;  
  
    // 填充结构体中的各成员  
    wc.hInstance = hInstance;  
    wc.lpszClassName = mainClassName;  
    wc.lpfnWndProc = mainWndProc;  
    wc.hbrBackground = (HBRUSH) GetStockObject(WHITE_BRUSH);  
    wc.style = CS_DBLCLKS;  
    wc.hIcon = NULL;  
    wc.hCursor = NULL;  
    wc.lpszMenuName = NULL;
```

```
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;

    // 注册窗口类
    if(!RegisterClass(&wc))
        return 0;
    return 1;
}

// 创建主窗口并显示
int showMainWnd(HINSTANCE hInstance, LPWSTR lpCmdLine, int nShowCmd)
{
    // 定义窗口句柄
    HWND hWnd;

    // 存储全局程序实例句柄
    hInst = hInstance;

    // 创建窗口
    hWnd = CreateWindow(
        mainClassName,
        TEXT("myFirst"),
        WS_VISIBLE,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        NULL,
        NULL,
        hInstance,
        NULL);

    // 如果窗口不存在, 返回 0 并退出
    if(!IsWindow(hWnd))
        return 0;

    // 显示并刷新窗口
    ShowWindow(hWnd, nShowCmd);
    UpdateWindow(hWnd);

    return 1;
}

// 退出主窗口, 即退出程序
```



```
int quitMainWnd(HINSTANCE hInstance,int nExitCode)
{
    return nExitCode;
}

// 主窗口过程体, 窗口过程必须是回调函数
LRESULT CALLBACK mainWndProc(HWND hWnd,UINT msgCode,
                              WPARAM wParam,LPARAM lParam)
{
    INT i;

    // 查找对应的消息 ID 号, 并执行相应的消息处理过程
    for(i = 0; i < dim(mainMessages); i++)
    {
        if(msgCode == mainMessages[i].uCode)
            return (*mainMessages[i].functionName)(hWnd,msgCode,wParam,lParam);
    }

    // 对于不在消息查找表中的消息, 调用 Windows CE 的缺省窗口过程
    return DefWindowProc(hWnd,msgCode,wParam,lParam);
}

// 主窗口收到 WM_CREATE 消息后的处理过程体
LRESULT goCreateMain(HWND hWnd,UINT msgCode, WPARAM wParam,LPARAM lParam)
{
    HWND wndCB;

    // 创建命令栏控件
    wndCB = CommandBar_Create(hInst,hWnd,IDC_CMDBAR);

    // 在命令栏上加入 Close 按钮
    CommandBar_AddAdornments(wndCB,0,0);
    return 0;
}

// 主窗口收到 WM_PAINT 消息后的处理过程体
LRESULT goPaintMain(HWND hWnd,UINT msgCode, WPARAM wParam,LPARAM lParam)
{
    // 定义画板结构体对象
    PAINTSTRUCT ps;

    // 定义矩形结构体对象
    RECT rect;
```