



Windows CE驱动开发

www.farsight.com.cn

第1节 Wince系统概述

I Windows CE的发展

- § 1996年, 微软发布了windows CE 1.0
- § Windows CE .NET 4.x , 5.0被广泛应用
- § 2007年, 微软发布了当前最新版windows CE 6.0

I Windows Embedded CE

组件化、实时操作系统和功能强大的工具 — 用于构建各种映像占用空间量少的嵌入式解决方案




1.1 操作系统特性

- } 32位操作系统，多线程，多任务。
- } 精简化的WIN32API。
- } 支持多种主流CPU。
- } 灵活的电源管理功能。
- } 良好的通信能力。
- } 完善的图形用户界面。
- } 功能齐全的网络功能。
- } 完全的实时操作系统。

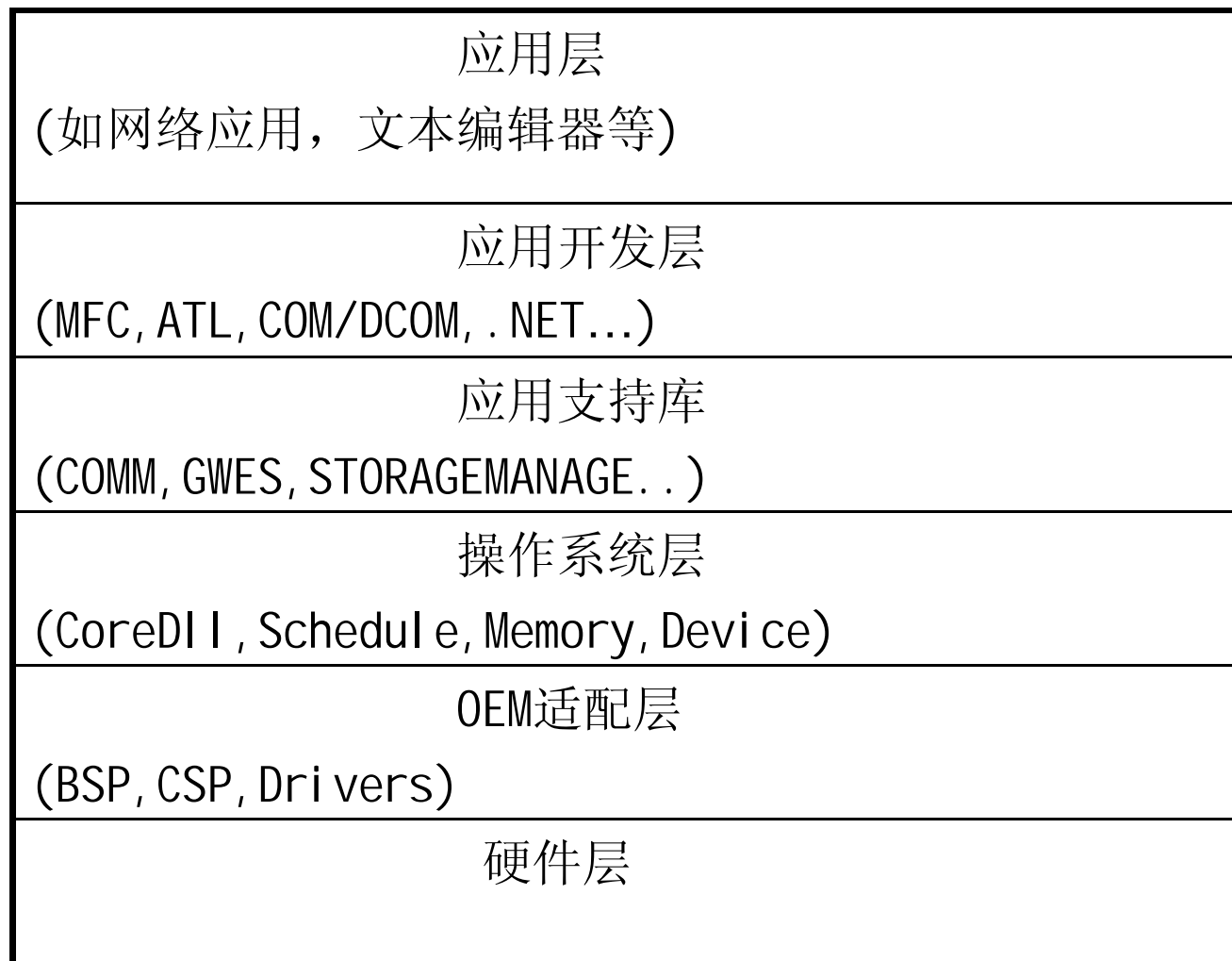


1.2 操作系统架构

采用了分层模型设计，使功能模块之间的调用关系更加清晰。更利于进行系统定制。

- } 硬件层
 - } OEM层
 - } 操作系统层
 - } 应用层
-
- 

操作系统分层模型



第2节 系统开发综述

- 2.1 设备开发流程。
- 2.2 开发工具的选择。
- 2.3 驱动开发



2.1 设备开发流程

- 1 硬件设计或选型。
- 2 选定BSP。
- 3 获得驱动程序。
- 4 裁减内核。
- 5 生成SDK。
- 6 编写应用软件。



2.2 开发工具的选择

- } 基于系统定制-选择Platform Builder。
 - } 基于应用程序开发-选择Embedded Visual C++。
 - } 在Wince6.0中,Platform Builder作为一个插件被集成到了VS2005中。VS2005可以进行系统定制开发以及应用程序开发。
 - } PB是进行 WINDOWS CE操作系统开发和定制的集成开发环境。
 - } PB提供了所有设计, 创建, 修改, 调试的工具。
-
- 

2.3 驱动开发



I 什么是设备驱动程序

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,大多数操作系统上的应用程序都是与硬件无关的,应用程序的开发者和最终用户通常都不必关心底层的硬件到底是如何工作的。**Wince**操作系统之所以能够支持各种各样的硬件设备,毫无疑问,是因为对每一款硬件设备,都有其对应的设备驱动程序。

设备驱动程序是操作系统与硬件交互的方式,是连接硬件与操作系统之间的桥梁。通常,设备驱动程序在操作系统与硬件之间扮演着特殊的角色,它们是一个个独立的“黑盒子”,使某个硬件可响应一些定义良好的编程接口。这组标准化的接口通常由操作系统定义,通常与驱动程序的类型相关而与具体的硬件无关。

I 何时编写设备驱动程序

- Ø 操作系统没有内置支持的硬件
- Ø 扩展已有的驱动功能
- Ø 提供软件层面的服务

I 编写驱动的要害

- Ø 要了解驱动程序所针对的硬件
- Ø 要了解目标操作系统下驱动的工作机制
- Ø 要了解驱动程序的模型

BSP包及驱动与系统的接口



构成元素	功能说明
Boot loader	完成将操作系统映像下载到设备存储器的功能
OEM adaptation layer	和操作系统内核相连接，完成硬件的初始化和管理工作。
Device drivers	对板载或者在系统运行中加载的外部设备提供驱动支持。
Configuration files	对BSP的环境变量进行设定

I 设备驱动程序控制硬件

I 对于某些硬件，是没有驱动的

∅ CPU

∅ 内存

I **Wince**驱动分类

∅ 本地设备驱动

∅ 分层驱动程序

∅ 流接口设备驱动


∅ 单体驱动程序



- 内建设备驱动程序用于低级、内置设备，提供一组定制的接口可通过移植、定制微软提供的驱动样例来实现。
- 内建驱动部分典型样例：
 - } 触摸屏驱动
 - } 显示驱动
 - } 鼠标及键盘驱动
 - } 打印机驱动

- } 大部分内建驱动程序采用分层结构。
- } PDD输出DDSI 函数接口，MDD通过这些函数接口来调用PDD层的功能函数。
- } MDD为操作系统提供用来访问MDD层的DDI 函数接口
- } MDD层一般以库文件形式存在，即DDSI 函数接口形式固定。

- **DDI(设备驱动接口)**是一类功能接口函数的统称，由分层驱动程序中的**MDD**层或者是单层驱动程序输出，由操作系统模块进行调用。
- **DDSI(设备驱动服务接口)**是一类由**PDD**层输出的接口函数，由**MDD**层进行调用。相同种类的设备驱动可以使用同样的**DDI**接口。

-
- Ø 操作系统通过DDI函数访问内建驱动的MDD层。
 - Ø MDD层以库形式存在，其向下调用的DDSI函数形式固定，但内容为空，需PDD层实现。
 - Ø PDD层以代码形式存在，需实现相应的DDSI函数内容。
-
- 


- } MDD层包含的代码在开发该类型的所有设备驱动时是通用的。
- } MDD层调用特定的PDD层程序来访问硬件或和硬件相关的信息。
- } 当使用分层驱动时，开发驱动时可以对微软提供的某些通用MDD层代码进行复用，而只需重新开发针对特定硬件环境的PDD 层代码。
- } 同样，如果想将某个驱动程序移植到新的硬件平台上，只需对PDD层代码进行移植，而MDD层代码则无需改动即可直接使用。

第3节 触摸屏驱动实例分析

- 3.1 驱动目的
- 3.2 驱动流程
- 3.3 驱动添加
- 3.4 驱动调试



3.1 驱动目的

- } 用户点击触摸屏时，将获取到的触摸屏坐标转换为实际的系统桌面坐标。
 - } 将用户动作(单击，双击，长点等)转化为相应的事件，通知GWES。
 - } 驱动调用方：GWES(图形界面系统)
-
- 

3.2 驱动流程

- } 用户点击触屏，触发中断。
 - } MDD层响应中断，调用驱动pdd层DDSI函数 TouchPanel Enable，启动触屏驱动。
 - } 接下来MDD层调用Ddsi TouchPanel GetDeviceCaps，来向获取当前触屏的信息。
 - } PDD层应根据查询的内容（通过对函数中由MDD层传递过来的参数iIndex判断）返回相应的数据(校正的频率，所需校正坐标的个数，校正基准点的坐标)。
-
- ▶

- } 针对当前获取的触屏点，调用Ddsi TouchPanel GetPoint。
- } 在这个函数中，会进行一系列算法校正，确保将触屏点转化为相应的桌面坐标。
- } 通过该函数的指针型参数，回传给MDD层。
- } tchmdd.lib为MDD层，负责根据PDD层回传的坐标点和事件信息，报告给GWES.exe。
- } MDD层代码未公开，用户只需修改PDD层代码，确保返回正确的坐标及事件参数即可。

3.3 驱动添加



- } 了解触摸屏驱动的接口，MDD层与PDD的接口，找到驱动的入口点。
- } 实现MDD层调用的PDD层代码。
- } 通过微软提供的公用源码库获取有用信息。
- } 公用源码库：
..\WINCE600\PUBLIC\COMMON\OAK\DRIVERS

驱动相关的配置文件

} Dirs、Sources、Platform.bib、Platform.reg等文件

举例:

```
RELEASETYPE=PLATFORM // 驱动生成文件的释放类型,
TARGETNAME=EduLed // 驱动生成目标文件的名字
DEFFILE=EduLed.def // 指定LED驱动的定义文件, 即上面提到的
*.def 文件
TARGETTYPE=DYNLINK // 驱动编译后生成库的类型, 这里为动态链接
库
DLLENTY=DIIMain // 驱动的入口函数
TARGETLIBS= \
    $_COMMONSDKROOT\lib\$_CPUINDPATH\coreDll.lib \ // 链接内核相关的
    库
SOURCELIBS= \
    $_PLATCOMMONLIB\$_CPUINDPATH\pxa27x_ms_v1_xllp.lib \ // 需要链
    接的源码库
SOURCES= \
```



3.4 驱动调试

- } 基于第三方的调试工具
- } 基于KITL调试工具
- } 串口调试方法



Wince系统定制与驱动开发班

- } 开发环境的建立
- } 操作系统的创建和调试
- } 内核修改
- } 板支持包的创建
- } 驱动程序开发



The success's road

www.farsight.com.cn

谢谢!

