

# 基于 Windows CE 的 OLED 驱动程序设计与实现

王星全, 兰明蛟, 郑振华

(总参谋部通信训练基地, 河北 张家口 075100)

**摘要:**介绍了 Windows CE 的体系结构和中断处理机制,研究了 Windows CE 设备驱动程序的类型和初始化过程,以维信诺 VGG13264C 132×64 OLED 显示模块的 WindowsCE 驱动程序设计为例,详细阐述了嵌入式 Windows CE 驱动程序的开发过程。

**关键词:** Windows CE; 驱动模型; OLED 驱动程序; 开发过程

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2011)20-0017-04

## Design and implementation of OLED drivers based on Windows CE

Wang Xingquan, Lan Mingjiao, Zheng Zhenhua

(General Staff Communications Training Base, Zhangjiakou 075100, China)

**Abstract:** Introduces the system structure of Windows CE and interrupt handle system, studies the types and the initialization process of Windows CE equipment drivers, it takes the Windows CE driver development of Visionox VGG13264C 132×64 OLED display module as an example, expounds the development process of device driver for the embedded Windows CE.

**Key words:** Windows CE; driver model; OLED driver; developing process

随着计算机技术和嵌入式技术的迅猛发展,嵌入式软、硬件设计需求越来越广泛,嵌入式软件开发,特别是嵌入式软件的驱动开发,成为一个热点。

Windows CE 是一个新的、可移植的、实时的、模块化的操作系统<sup>[1]</sup>,具有简捷、高效的完全抢先式多任务操作核心,支持强大的通信和图形显示功能,能够适应广泛的系统需求,在最新一代的工业和家用电子设备中得到了广泛的应用。

在进行嵌入式 Windows CE 项目开发的过程中,经常需要根据需求定制显示屏,如何为定制的显示屏开发 Windows CE 驱动程序成为嵌入式系统开发的关键步骤。本文在深入研究 Windows CE 驱动程序工作原理的基础上,以维信诺 VGG13264C 132×64 OLED 显示模块的 WindowsCE 驱动程序设计为例,详细阐述了嵌入式 Windows CE 驱动程序的开发过程,并对设计实现的驱动程序进行了测试。

### 1 Windows CE 的系统体系结构

Windows CE 是微软设计开发的一种可升级的抢先式多任务 32 位实时操作系统。一个基于 Windows CE 的平台主要由以下几部分组成:Windows CE 内核组件、OAL 层和驱动程序。OAL 层和驱动程序作为目标设备和

Windows CE 内核组件之间的接口,将操作系统上层与硬件隔离,便于支持硬件扩展和即插即用功能<sup>[2]</sup>,图 1 详细描述了三者之间的关系。



图 1 基于 Windows CE 的目标平台及其组件

### 2 Windows CE 中断处理

Windows CE 将中断处理分为两个步骤:中断服务例程(ISR)和中断服务线程(IST)<sup>[3]</sup>。把每个硬件的设备中断请求(IRQ)和一个 ISR 联系起来,当一个中断发生并

未被屏蔽时,内核调用该中断注册的 ISR。因为 ISR 运行于内核模式,所以应设计得尽可能短,ISR 引导内核调度和启动合适的 IST,IST 在设备驱动程序软件模块中编写,运行于用户模式,它从硬件获取或向硬件发送数据和控制代码,并进一步处理设备中断。Windows CE 还提供了中断嵌套功能,即在一个 ISR 运行时,内核并不关闭中断,当优先级比之高的中断发生时,内核保存当前执行的中断服务例程 ISR 的运行状态,挂起该 ISR,转而执行更高优先级中断的 ISR。等到高优先级中断的 ISR 执行完后,被挂起的低优先级中断的 ISR 才重新开始被 CPU 调度执行。Windows CE 的中断处理框图如图 2 所示。

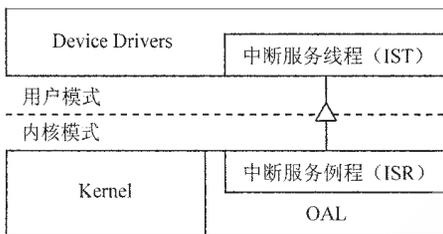


图 2 Windows CE 的中断处理框图

### 3 Windows CE 设备驱动程序

设备驱动程序将操作系统和设备连接起来,使操作系统能够识别设备,并为应用程序提供服务。

#### 3.1 设备驱动程序模型

Windows CE 提供了用于驱动开发的模型<sup>[4]</sup>,其中包括来自其他操作系统的驱动程序模型。

因为有这些多变的驱动程序模型,使 Windows CE 可以适应大部分的内部和外围设备。目前,Windows CE 提供了 4 种设备模型,其中两种是专用于 Windows CE 模型,另外两种外部模型来自其他的操作系统,汇总如图 3 所示。

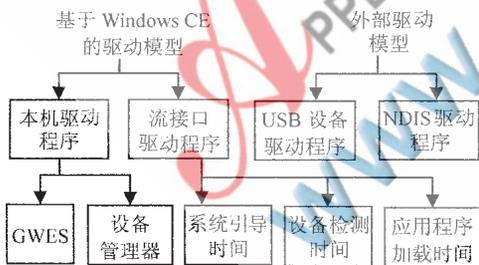


图 3 驱动程序模型

#### 3.2 本机驱动程序

要把 Windows CE 移植到目标平台上,必须为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,所以称这类驱动程序为本机驱动程序。Windows CE 平台生成器提供本机设备驱动程序的样本,可考虑把本机设备驱动程序样本应用到平台上,而不需要再从头开发自己的

本机驱动程序。用户可根据驱动程序样本快速开发自己的驱动程序。

#### 3.3 流接口驱动程序

流接口驱动程序由设备管理应用程序加载、管理和卸载,与具有单独目的接口的本机驱动程序相比,所有流接口驱动程序使用同一个接口并调用同一个函数集——流接口函数。流接口驱动程序把外设抽象成一个文件,使用标准文件 I/O 函数和电源管理函数,这些函数由 Windows CE 操作系统的内核使用。流接口驱动程序的入口点函数为:XXX\_Init、XXX\_Deinit、XXX\_IOControl、XXX\_Open、XXX\_Close、XXX\_PowerDown、XXX\_PowerUp、XXX\_Read、XXX\_Seek、XXX\_Write。流接口驱动程序函数的作用及调用关系如表 1 所示。

表 1 Windows CE 下流接口驱动程序函数作用及调用关系

函数名	功能表述	调用关系
XXX_Init	设备初始化	由设备管理器调用
XXX_Open	打开设备	通过 CreateFile 调用
XXX_Read	从设备读取数据	通过 ReadFile 调用
XXX_IOControl	向设备发出命令	通过 DeviceIoControl 调用
XXX_Write	向设备写入数据	通过 WriteFile 调用
XXX_Seek	移动数据指针	通过 SetFilePointer 调用
XXX_PowerUp	给设备上电	
XXX_PowerDown	给设备断电	
XXX_Close	关闭设备	通过 CloseHandle 调用
XXX_Deinit	设备卸载	由设备管理器调用

流接口驱动程序从设备管理器和通过文件系统调用的应用程序接受命令,装入驱动将这些命令翻译成其所控制设备的适当操作的所有信息。所有的流接口驱动,无论它管理的是内置式设备还是可安装式设备,是系统引导加载还是动态加载,它们与其他的系统组件都有类似的交互过程。

#### 3.4 设备驱动程序的初始化过程

在 Windows CE 中,所有的设备驱动程序都以动态链接库(DLL)的形式存在,需要通过进程加载。设备驱动程序向外提供接口函数,应用程序使用 API 对设备进行访问。Windows CE 共有三类系统进程用来加载驱动程序:Device.exe、GWES.exe、FileSys.exe。驱动程序的初始化过程为:设备上电——启动 bootloader——启动 NK.exe——启动 Device.exe——初始化数据结构和 I/O——加载总线枚举器——枚举注册表 Driver/BuildIn 下所有的子键。这里的枚举就是循环调用 ActivateDeviceEx 函数加载驱动程序。

### 4 Windows CE 开发流接口驱动程序实例

以维信诺 VGG13264C 132×64 OLED 显示模块的驱动程序为例,具体阐述 Windows CE 驱动开发的过程。

## 4.1 硬件结构

系统硬件以飞凌 OK6410-B 开发板为主板,以维信诺 VGG13264C 显示模块为 OLED 显示屏。系统硬件框图如图 4 所示。

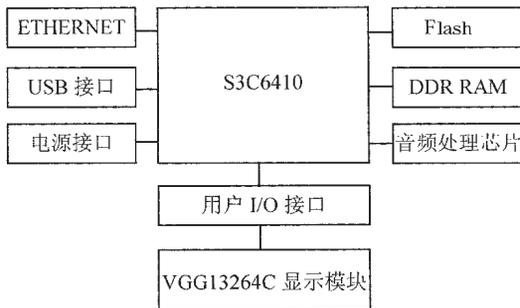


图 4 系统硬件结构框图

飞凌 OK6410-B 开发板基于三星公司最新的 ARM11 处理器 S3C6410,拥有强大的内部资源和视频处理能力,可稳定运行在 667 MHz 主频以上,支持 Mobile DDR 和多种 NAND Flash。FL6410 开发板上集成了多种高端接口,如复合视频信号、摄像头、USB、SD 卡、液晶屏、以太网等,丰富的接口可帮助用户实现高端产品级设计。

维信诺 VGG13264C 是 132 列×64 行点阵的 OLED 单色、字符、图形显示模块,使用单芯片 SSD1303T6 进行驱动,具有 8 位并行数据接口,模块内含 132×64 显示数据 RAM,通过用户 I/O 接口连接到飞凌 OK6410-B 开发板。

## 4.2 流接口驱动程序开发过程

Windows CE 提供了两种编写流接口驱动的方法:

(1)通过编写动态链接库的方法实现;(2)通过修改 BSP 的方法实现。由于第二种方法可以加快开发进度,在该实例中,使用修改 BSP 的方法进行流接口驱动程序开发,其开发步骤为:首先在 BSP 中新建一个目录,编写驱动程序文件并确定驱动程序向外提供的接口函数,然后进行注册表设置,最后进行测试。

## (1)驱动程序代码编写

设备初始化使用 OED\_Init 函数实现,设备卸载使用 OED\_Deinit 函数实现。在 OED\_Init 函数中,使用 DrvLib\_MapIoSpace 函数实现物理地址到虚拟地址映射;在 OED\_Deinit 函数中,使用 VirtualFree 函数释放驱动程序使用的虚拟内存从而实现了设备的卸载。

设备的 I/O 控制向设备发出命令,按照维信诺 VGG13264C OLED 显示模块的时序关系,通过控制相应管脚的电平高低实现向设备发出命令。设备的 I/O 控制在 OED\_IOControl 函数中实现,函数部分代码如下:

```
switch (dwIoControlCode) {
    case IOCTL_OLED_INIT:
        v_pIOPregs->GPPCON&=0xF000FFFF;
        v_pIOPregs->GPPCON|=0x5550000;
        .....
```

```
Delay_ns(2);
v_pIOPregs->GPEDAT|=(0x01<<4);
Delay_ns(2);
v_pIOPregs->GPEDAT|=(0x01<<2);
```

```
break;
```

```
case IOCTL_OLED_WRITECOMMAND:
```

```
v_pIOPregs->GPEDAT&=~(0x01); //A0 低
v_pIOPregs->GPEDAT|=(0x01<<3); //CS 高
v_pIOPregs->GPEDAT|=(0x01<<1); //WR 高
Delay_ns(1);
v_pIOPregs->GPEDAT&=~(0x01<<3); //CS 低
v_pIOPregs->GPEDAT&=~(0x01<<1); //WR 低
if((0x01<<7)&(*pInBuf))
```

```
v_pIOPregs->GPQDAT|=(0x01<<6);
```

```
else v_pIOPregs->GPQDAT&=~(0x01<<6);
```

```
.....
```

```
Delay_ns(1);
```

```
v_pIOPregs->GPEDAT|=(0x01<<1); //WR 高
```

```
Delay_ns(1);
```

```
v_pIOPregs->GPEDAT|=(0x01<<3); //CS 高
```

```
break;
```

```
case IOCTL_OLED_WRITEDATA:
```

```
v_pIOPregs->GPEDAT|=(0x01); //A0 高
```

```
v_pIOPregs->GPEDAT|=(0x01<<3); //CS 高
```

```
v_pIOPregs->GPEDAT|=(0x01<<1); //WR 高
```

```
v_pIOPregs->GPEDAT&=~(0x01<<3); //CS 低
```

```
v_pIOPregs->GPEDAT&=~(0x01<<1); //WR 低
```

```
if((0x01<<7)&(*pInBuf))
```

```
v_pIOPregs->GPQDAT|=(0x01<<6);
```

```
else v_pIOPregs->GPQDAT&=~(0x01<<6);
```

```
.....
```

```
Delay_ns(1);
```

```
v_pIOPregs->GPEDAT|=(0x01<<1); //WR 高
```

```
Delay_ns(1);
```

```
v_pIOPregs->GPEDAT|=(0x01<<3); //CS 高
```

```
break;
```

```
}
```

## (2)向外提供函数接口

通过配置 OED.def 文件导出 OLED 提供的接口函数,供设备管理器使用。在 OED.def 文件中添加如下代码:

```
LIBRARY OED
EXPORTS OED_Init
        OED_Deinit
        OED_Open
        OED_Close
        OED_Read
        OED_Write
        OED_Seek
```

OED \_PowerDown

OED \_PowerUp

OED \_IOControl

### (3)注册表配置

注册表文件 Platform. reg 用来配置与 OLED 驱动有关的信息,配置如下:

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\IoControler]
```

```
“Prefix”=“OED”
```

```
“Dll”=“OED.dll”
```

```
“Index”=dword:0
```

随着嵌入式系统广泛应用到信息家电、工业控制、移动通信、互联网等领域,大量嵌入式系统的开发必将对人类科技的发展起着重要的作用。本文介绍的基于嵌入式操作系统 Windows CE 的驱动程序会帮助读者开发出更多的嵌入式系统产品。

### 参考文献

[1] GOGGIN T A. Windows CE 高级开发指南[M]. 尤滔,

张平,周晓权,译,北京:电子工业出版社,2001.

[2] Microsoft . Microsoft Windows CE Device Driver Kit[M]. 北京:北京希望电子出版社,2000.

[3] 胡军辉,王友钊.Windows CE 设备驱动程序开发[J].计算机工程,2006,32(16):41-43.

[4] 李长河,王永强,刘刚,等.嵌入式 Windows CE 设备驱动的研究实现[J].微机发展,2003,13(7):69-72.

(收稿日期:2011-07-18)

### 作者简介:

王星全,男,1959年生,高级工程师,主要研究方向:军事通信学、军队指挥学。

兰明蛟,男,1978年生,硕士,讲师,主要研究方向:军事通信学。

郑振华,男,1976年生,讲师,主要研究方向:电子与信息工程。

电子技术应用  
APPLICATION OF ELECTRONIC TECHNIQUE  
www.ChinaAET.com