

首页 > 技术 > 设计应用

视频采集卡Windows95驱动程序设计

冯德锦

诺基亚(中国)投资有限公司研究开发中心(100013)

2008-11-17

摘要: 介绍了Windows95系统中视频采集卡驱动程序的原理和结构,较详细地介绍了虚拟设备驱动程序和Video For Window视频采集驱动程序设计技术。

关键词: 视频采集 驱动程序 Windows95

在计算机视频应用中,视频采集卡承担着将模拟视频信号转换成数字视频信号的任务。Windows95是目前应用比较广泛的操作系统,许多视频应用系统是基于该平台的。不但采集卡设计人员需要详细了解Windows95系统中视频采集卡驱动程序设计技术,应用程序设计人员也需要理解驱动程序的工作原理,以便设计出性能更好的应用程序。本文以PCI总线视频采集卡为例,较详细地介绍了Windows95系统中视频采集卡驱动程序设计技术。

1 Windows95驱动程序原理

Windows95作为多线程抢占式多任务操作系统,对硬件设备进行了全面而完善的管理。其设备驱动程序的主要目的是使系统中的各个任务共享硬件设备,在任务切换时确保硬件设备的一致性。Windows95设备驱动程序是分层设计的,层间定义了统一的接口,提高了程序的兼容性,便于系统升级和移植。系统将各类硬件设备的功能、行为进行归类整理,将实际设备抽象成逻辑设备。逻辑设备具有某一类型设备所共有的属性,具有统一的行为,并以一致的方式进行操纵。系统提供必要的函数和方法供应用程序操纵某一类型的逻辑设备。应用程序被严格禁止进行直接的硬件设备操作,所需的操作应当调用相应的系统API函数来完成。视频采集卡的设备驱动程序属于MultiMedia类中的VideoCap子类,由虚拟设备驱动程序CAPTURE.VXD和可安装设备驱动程序CAPTURE.DRV构成。其结构框图见图1。

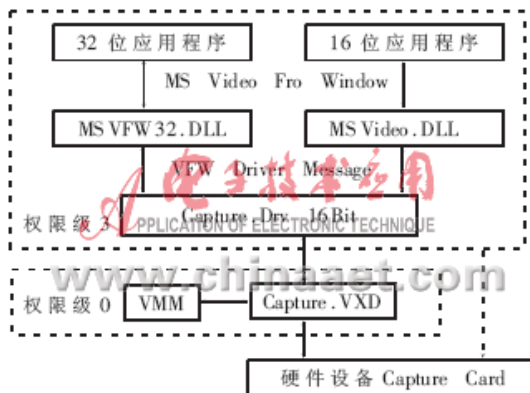


图1 Win95 视频采集设备驱动程序结构

虚拟机管理器VMM构成了Windows95系统的核心。它是一个32位保护模式操作系统,它的主要责任是创建、运行、监视以及终止虚拟机。虚拟机是一个可执行的任务,它由内存空间、CPU寄存器组、应用程序以及应用程序的支持软件如ROM BIOS、MS-DOS等组成,系统可以同时拥有多个虚拟机。当某个硬件设备的状态可能受到任务切换的干扰时,该设备就需要相应的虚拟设备驱动程序的支持。虚拟机管理器和虚拟设备驱动程序一同管理系统的软硬件资源,响应中断,进行I/O操作,确保多个应用程序共享资源,互不干扰。

Windows95视频采集卡设备驱动程序包括虚拟设备驱动程序和可安装设备驱动程序两部分,一般可以采用Microsoft Windows95 DDK (Device Driver Kit)进行设计。DDK中提供了进行设计所必要的文档、底层函数原形、输入库以及一些样例代码,但是未提供编程所需要的编译、汇编和连接程序。采用Visual C++ 5.0、Visual C++ 1.5以及MASM 6.0进行程序设计。设备驱动程序设计的好坏与采集卡的功能、性能、兼容性及稳定性密切相关。为保证驱动程序的质量,应当严格遵照Windows95 DDK的文档进行程序设计。

2 虚拟设备驱动程序设计

在视频采集卡的驱动程序中,虚拟设备驱动程序CAPTURE.VXD负责参与系统即插即用配置过程,管理视频采集卡的硬件设备,并向运行在权限级3的上层程序提供调用接口:一是将分配的资源返回,如内存选择子、I/O地址、中断号等,使得上层驱动程序可以使用这些资源来操纵采集卡硬件设备;二是完成一些在权限级3所不能进行的操作,如分配DMA缓冲区、查看内存映射页表等。虚拟机管理

热点专题

- 2008嵌入式技术创新及应用高峰论坛
- 2008飞思卡尔技术论坛
- Altera公司SOPCWorld 2008专题报道
- 第十届高交会电子展
- 科技闪耀北京奥运
- ADLINK DAY—2008年量测与自动化技术国际高峰论坛
- 中国电子学会Xilinx杯开放源码硬件创新大赛
- 赛灵思公司Virtex-5系列FPGA
- 3G知识
- IPTV
- 触摸屏技术
- RoHS

杂志精华

- 基于CC2430的无线传感器...
- 无线传感器网络应用系统综述
- 无线传感器网络在野外测量中的...
- 基于竞争的无线传感器网络
- 用于矿井环境监测的无线传感器...
- 具有自适应通信能力的无线传感...
- 基于传感器网络技术的深孔测径...
- 基于无线传感器网络的家庭安防...
- 基于ATmega128L与C...
- 无线传感器网络中移动节点设备...

器和虚拟设备驱动程序运行在一个单一的32位平板模式地址空间中,权限级为0。系统建立两个基地址为0、界限为4GB的全局描述符选择子分别用于代码段和数据段,程序不应当直接修改段寄存器、内存描述符表、中断描述符表。

2.1 设备声明

虚拟设备驱动程序必须包含一个声明,它一般位于程序的前部,声明中包括虚拟设备驱动程序的名称、版本号、调用接口等一些重要信息。声明由宏Declare_Virtual_Device完成,语法如下:

```
Declare_Virtual_Device <设备名称>, <主版本号>, <次版本号>, <设备控制过程>, <设备标识符>, <初始化次序>,,
```

设备名称定义了虚拟设备驱动程序的名称。设备控制过程接受来自虚拟机管理器的消息,并根据不同消息进行不同的操作。设备标识符是一个16位整数,它是该程序在系统中的唯一标识。初始化次序确定虚拟设备驱动程序的装载顺序,首先装入初始化次序值较小的虚拟设备驱动程序。V86 API接口用于接受来自虚拟8086模式程序的调用。PM API接口接受来自16位保护模式程序的调用。

2.2 设备控制过程

虚拟机管理器在系统的运行状态发生改变时向设备控制过程发送系统控制消息(System Control Message),以便虚拟设备驱动程序能够跟踪系统状态,进行必要的处理。发送消息的时机包括:系统初始化、系统退出、虚拟机状态改变、虚拟设备驱动程序初始化等。一个设备控制过程的例子如下:

```
BeginProc WBD848_Control
```

```
Control_Dispatch Sys_Dynamic_Device_Init, WBD848_Dyn_
Device_Init
```

```
Control_Dispatch Sys_Dynamic_Device_Exit, WBD848_Dyn_
Device_Exit
```

```
Control_Dispatch PnP_New_DevNode, WBD848_PnP_
New_DevNode
```

```
Control_Dispatch W32_DEVICEIOCONTROL, WBD848_W32_
DeviceIOControl
```

```
Clc
```

```
ret
```

```
EndProc WBD848_Control
```

与视频采集卡即插即用配置过程相关的消息是PnP_New_DevNode。系统首先检测视频采集卡硬件的存在,然后根据PCI接口板厂商和器件标识在注册表中找到对应的虚拟设备驱动程序并装入内存,随后系统向该虚拟设备驱动程序发送PnP_New_DevNode消息。因为视频采集卡属于多媒体设备,设备驱动程序的装入应由MMDEVLDR.VXD完成,所以虚拟设备驱动程序在处理PnP_New_DevNode消息时,调用MMDEVLDR.VXD的服务MMDEVLDR_Register_Device_Driver注册了回调函数PnP_Config_Handler,系统根据采集卡PCI配置空间为其分配资源后将调用此回调函数。在PnP_Config_Handler中可以使用CM_Get_Alloc_Log_Conf服务获得系统为采集卡分配的资源。处理系统控制消息PnP_New_DevNode的例程如下:

```
BeginProc WBD848_PnP_New_DevNode
```

```
mov eax, ebx ;虚拟机句柄
```

```
mov ebx, offset32 PnP_Config_Handler ;即插即用处理例程
```

```
VxDCall MMDEVLDR_Register_Device_Driver
```

```
mov eax, CR_SUCCESS
```

```
stc
```

```
ret
```

```
EndProc WBD848_PnP_New_DevNode
```

在回调函数PnP_Config_Handler中获得的资源包括内存、I/O、中断和DMA四种。PCI总线视频采集卡主要使用内存和中断资源,它包括一个内存窗口作为内存映射I/O和一个中断IRQ号。虚拟设备驱动程序应当使用_MapPhysToLinear服务将内存窗口物理地址映射到线性地址空间中,并使用_Allocate_GDT_Selector服务分配内存选择子,以便16位应用程序访问该内存区域。

2.3 应用程序接口

虚拟设备驱动程序为虚拟8086模式、16位及32位保护模式应用程序分别提供应用程序编程接口,使运行在权限级3上的应用程序可以访问虚拟设备驱动程序。前两种接口在设备声明时定义,应用程序通过中断调用INT28H AX=1648H获得虚拟设备驱动程序的入口地址。32位接口通过W32_DEVICEIOCONTROL系统控制消息实现,应用程序则使用系统API函数DeviceIOControl对虚拟设备驱动程序进行调用。

当16位应用程序对入口地址发出远调用后,虚拟机管理器将应用程序的寄存器内容保存在Client_Reg_Struct结构中,并将指向该结构的指针赋予EBP寄存器,然后调用相应的虚拟设备驱动程序。虚拟设备驱动程序必须访问Client_Reg_Struct结构中的数据以获得应用程序的调用参数。一般利用AX寄存器传递功能号,利用其它寄存器传递参数,执行结果通过Client_Reg_Struct结构返回。16位保护模式应用程序使用选择子——偏移量地址模型,而虚拟设备驱动程序使用32位平板式地址模型,如果参数是指针的形式传递给虚拟设备驱动程序,需要使用宏Client_Ptr_Flat进行转换。

本采集卡中虚拟设备驱动程序提供的主要功能是:(1)使应用程序获得系统为采集卡分配的资源,如内存映射、I/O地址和中断IRQ号,以便操纵采集卡;(2)提供有关内存页表的信息,以便进行DMA操作。16位应用程序接口的部分代码如下:

```
;***** Copy Page Table *****
```

```
Client_Ptr_Flat eax, ES, BX ;es: bx指向页表缓冲区
```

```
Client_Ptr_Flat ebx, DI, SI ;di: si指向DMA缓冲区
```

```
shr ebx, 12 ;起始页号
```

```
movz ecx, [ebp].Client_CX ;cx为需要拷贝的页表项个数
```

```
VMMcall _CopyPageTable
```

```
mov [ebp].Client_EAX, eax ;ax返回执行结果
```

```
ret
```

```
;***** Get Resource *****
```

```
movzx eax, _g_wSelector ;存储器映射I/O地址选择子
```

```

mov     [ebp].Client_EAX, eax
movzx   eax, _g_wIRQ          ;IRQ号
mov     [ebp].Client_EBX, eax
ret

```

32位应用程序接口的功能与16位接口十分相似,只是将存储器映射I/O地址以32位线性地址的方式返回,以适应32位平板模式的寻址要求。另外32位接口不返回中断IRQ号,因为在32位应用程序中不进行有关中断的操作。

3 可安装设备驱动程序

可安装设备驱动程序实质上是一个16位动态链接库,可以按照一般的动态链接库设计方法进行设计。它具有入口函数DriverProc,用于接收系统发送的消息。

3.1 Video For Window编程模型

视频采集卡可安装设备驱动程序采用Video For Window程序模型,它从各种各样的视频采集卡抽象出一个统一的逻辑结构,以便上层程序调用。上层程序主要是指由AVI Cap32.dll等系统动态链接库构成的视频采集引擎。视频采集引擎负责显示视频数据、分配数据缓冲区、操纵系统调色板、存储AVI文件等工作,它们将应用程序的高级调用翻译成低级消息发送给可安装设备驱动程序。可安装设备驱动程序在消息的控制下,操纵采集卡硬件设备,完成视频数据的采集。Video For Window驱动程序逻辑结构如图2所示。

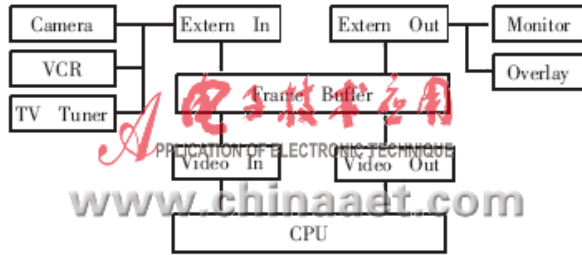


图2 Video For Window 驱动程序逻辑结构

Video For Window驱动程序模型包括视频源、帧存储器、显示设备、CPU和四个视频数据通道。帧存储器只具有逻辑上的意义,可能在采集卡上或位于主机内存中或者根本不存在。它在图2中只表明处于该位置的应当是已经解码完整的视频数据。四个视频数据逻辑通道的功能如下:

External In代表视频信号由模拟向数字的转换过程,包括采样量化、解码等工作。该通道负责视频信号的选择(摄像机、录像机、调谐器等),视频信号制式选择(NTSC、PAL、SECAM等),亮度、对比度、色调的调节等工作。

Video In 代表视频数据由帧存储器向系统缓冲区的传输过程。该通道负责视频图像数据的传输工作,如设定图像大小、彩色格式,确定图像帧的同步定时等。

External Out 代表视频数据向显示设备的直接传输过程。该通道负责视频图像的实时显示工作,如确定显示窗口位置和数据格式,进行显示窗口的剪裁等。

Video Out 代表视频数据由系统缓冲区向帧存储器的反向传输过程。该通道负责视频数据的回放,具有视频解压缩功能的采集卡可能需要这种反向的传输,以便对压缩数据进行解码。

一个最简单的视频采集驱动程序至少应当实现External In和Video In两路数据流,也就是实现视频信号的采集、解码和向主机内存的传输。通常PCI总线视频采集卡还可以实现External Out数据流,即实现视频数据的实时显示。

3.2 数据传输

驱动程序将采集到的数据传送到上层程序时使用VIDEOHDR结构,该结构包含了数据缓冲区、数据长度、时间戳等信息,定义如下:

```

typedef struct {
LPSTR lpData;          //address of video buffer
DWORD dwBufferLength;  //size, in bytes, of the data buffer
DWORD dwBytesUsed;
DWORD dwTimeCaptured;
DWORD dwUser;          // user-specific data
DWORD dwFlags;
DWORD dwReserved[4];  // reserved; do not use
}VIDEOHDR;

lpData          视频数据缓冲区指针,由采集引擎填写;
dwBufferLength  视频数据缓冲区长度,由采集引擎填写;
dwBytesUsed     缓冲区中图像数据的实际长度,由驱动程序填写;
dwTimeCaptured  时间戳,从采集第一帧起以ms为单位记录,用于视频数据的同步,由驱动程序填写;
dwFlags        标志,驱动程序应当填写VHDR_DONE以表示采集完成,如果此帧为关键帧还可以同时使用
                VHDR_KEYFRAME标志。

```

缓冲区通常由视频采集引擎分配,并将VIDEOHDR结构指针以消息参数形式发送给驱动程序。当进行单帧采集时,驱动程序每收到一个DVM_FRAME消息便采集一帧图像,驱动程序从DVM_FRAME消息返回时将VIDEOHDR结构返回给采集引擎。

序列图像的采集比较复杂,驱动程序与采集引擎间以缓冲区队列的形式进行数据交换,确保了连续的实时采集。采集引擎使用DVM_ADDBUFFER消息将空数据缓冲区不断地加到缓冲区队列的队尾。同时,驱动程序将空数据缓冲区从队列的队首取出,并进行视频采集。驱动程序将填充视频数据的缓冲区通过回调函数返回给采集引擎,由它进行显示、存盘等操作。添加缓冲区和取出缓冲区是两个相对独立的异步操作。采集引擎是缓冲区的生产者,它可以在系统空闲时分配若干个缓冲区并加入队尾。驱动程序是缓冲区的消费者,在采集卡硬件中断的驱动下它不断取出空缓冲区,并将用过的缓冲区返还给采集引擎。通过使用缓冲区队列提高了系统的数

据吞吐能力,确保了采集的实时性。

3.3 视频实时显示

视频图像实时显示功能又称为Overlay,它将视频图像以窗口方式在计算机屏幕上实时显示,显示的画面具有良好的视觉效果,显示窗口与Windows95的图形界面配合默契。要实现这些功能,首先采集卡必须具有将数据直接传输至显示卡的硬件设备,因为实时显示数据传输量巨大,主机CPU是不可能承担这样的任务的;其次要求软硬件设备默契配合,以便恰当处理显示窗口的大小、定位及剪裁。这可以借助DirectDraw中的Overlay功能实现。

DirectDraw是一种软件接口标准,提供了对显示设备的直接访问。它将显示画面分成表面(Surface),每种表面具有不同的属性,如尺寸大小、彩色格式等。整个屏幕对应的表面称为主表面,程序可以在显存或内存中建立其它表面,并在这些表面中绘制图形,然后通过某种操作将这些表面的内容显示在主表面上,这些操作是高效率的,程序可以从中获得很高的性能。Overlay表面具有特殊的性质,它位于显存中,并且可以覆盖显示在主表面的任何位置。为了防止Overlay表面遮盖主表面中的某些重要部分如菜单、对话框等,Overlay表面还设有色键功能,只有主表面中的颜色值与色键值相同时,Overlay表面才覆盖主表面。

当驱动程序的External Out通道收到DVM_STREAM_INIT消息时,开始视频实时显示。首先启动DirectDraw接口,创建主表面和Overlay表面,并记录有关Overlay表面的信息,如尺寸、数据格式、内存地址等,同时为Overlay表面选择一个颜色作为色键,通常是洋红色。然后根据Overlay表面的信息初始化硬件设备,使DMA设备将视频数据直接送入Overlay表面所在的显存中。当显示窗口移动或改变大小时,视频采集引擎会发送DVM_UPDATE消息通知驱动程序更新显示,驱动程序应当在视频窗口中填充色键的颜色,并移动Overlay表面使其与视频窗口对齐。因为可安装设备驱动程序是16位模块,而DirectDraw位于32位模块中,在采集卡驱动程序中不能直接调用DirectDraw接口,故需借助Flat Thunk在16位与32位模块间建立联系。

本设备驱动程序充分利用PCI总线的带宽和Windows95系统资源,获得了较高的采集速率(384×288×24bits 25fps),实现了实时视频显示。驱动程序性能稳定,兼容性良好,在运行Windows95的586以上系统中可稳定运行,并可运行于Windows98上。

参考文献

- 1 Microsoft Corporation. Microsoft Windows95 Device Driver Kit. 1997
- 2 曹家智,王蓉,刘乃琦. 80486/80386系统设计和应用. 成都: 电子科技大学出版社, 1992

在线联系

[添加到收藏夹](#)

关于“[视频采集卡Windows95驱动程序设计](#)”, 我有如下需求或意向:

用户名: 密码: 验证码:  [欢迎注册](#)

相关应用

- 基于StrongARM的视频采集与处理系统
- 基于DSP的USB口数据采集分析系统
- 基于USB总线的实时数据采集系统设计与实现
- 基于PCI的高速媒体网关系统及其驱动程序设计
- 基于PCI总线和DSP芯片的图像处理平台的硬件设计
- Windows CE中实现蓝牙串口驱动程序

[版权声明](#) | [投稿须知](#) | [《电子技术应用》投稿](#) | [网站地图](#) | [帮助中心](#) | [广告中心](#) | [关于我们](#) | [管理员信箱](#)

[回到顶端](#)

《电子技术应用》编辑部版权所有

地址: 北京海淀区清华东路25号电子六所大厦

联系电话: 82306084 / 82306085 传真: 62311179 京ICP备05053646号

推荐分辨率1024*768 IE6.0版本

