

WinDriver

快速入门指南

五分钟学会写设备驱动程序。



谁必须用 WinDriver?

1. 硬件开发者--用这个驱动程序向导来快速测试你的新硬件。
2. 软件开发者--用这个驱动程序向导来产生设备驱动代码驱动你的硬件。用 WinDriver 工具来检测和调试你的驱动程序。

WinDriver 支持哪个操作系统?

1. WinDriver 现在支持 Windows 95, 98, NT, 2000, Windows CE, Linux, Solaris 和 VxWorks. 检查一下 RFTech 网址看看在操作系统支持上是否有新的升级。
2. 利用 WinDriver,你写一样的驱动程序将会运行在以上所有的平台上--只要重新编译即可运行。学习 WinDriver 的方式, 请看 WinDriver 技术白皮书。

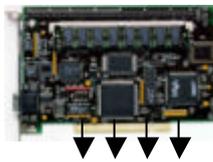
在哪获得更多深入的资料?

1. 要想得到 WinDriver 技术白皮书、更深入的资料和价值 30 天的免费使用, 请看 KRFtech 网址：
<http://www.krfttech.com/download.html>
2. 完整的 WinDriver 指南可以从 <http://www.krfttech.com/manuals.html> 下载。

六步建立设备驱动程序：

1 安装：

- 把硬件装置插入到 PC 机上
- 安装 WinDriver。



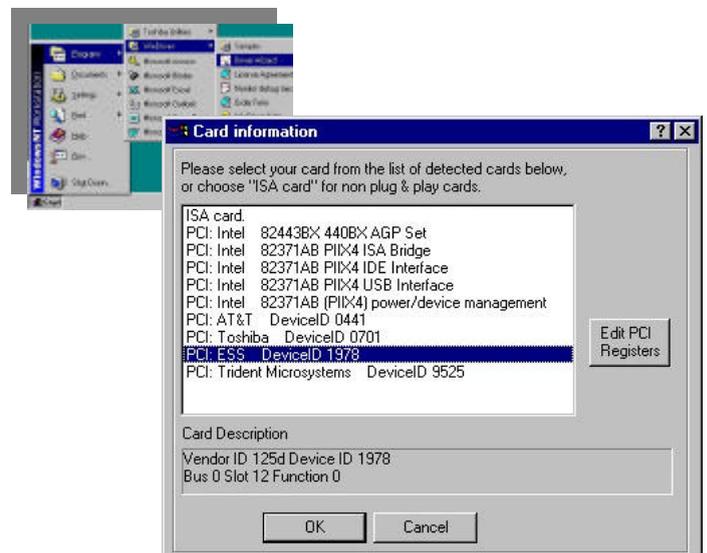
PCI/ ISA 总线

2 选择硬件：

- 从开始菜单里, 选择 WinDriver|Driver Wizard, 运行 DriverWizard 程序。

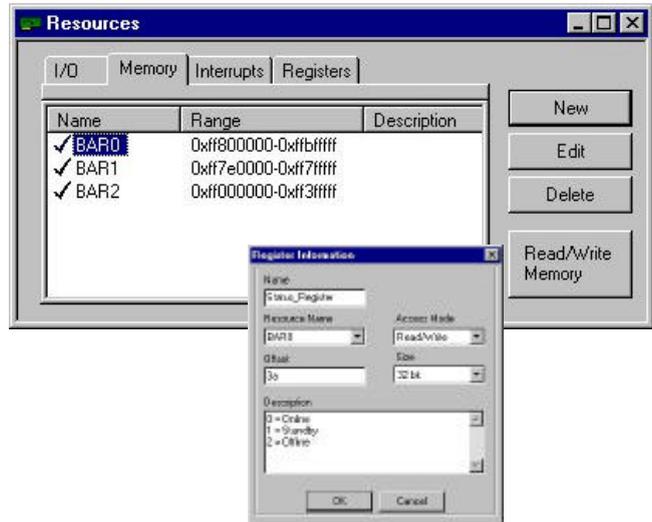
DriverWizard 将会在你的机器上显示所有的即插即用设备。

- 对于即插即用设备: 从列表中选择你的硬件设备。
- 对于非即插即用设备: 从列表中选择“ISA”(你的鼠标, 键盘, 内存等都是 ISA 设备)。



3 检测 / 定义硬件:

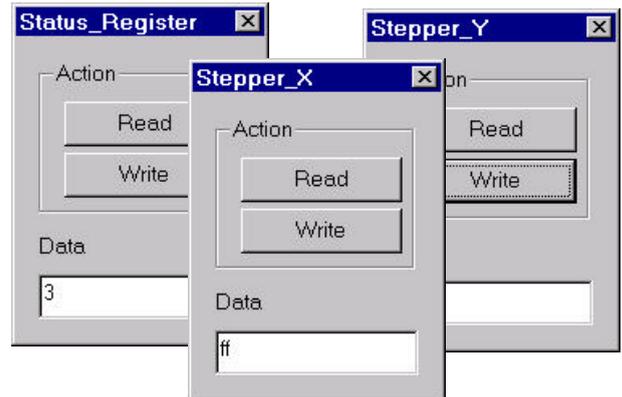
- Driver Wizard 将自动测试即插即用设备资源(I/O 范围, 内存范围以及中断)。你可以手动定义寄存器。
- 对于非即插即用设备: 你需要手动定义硬件资源。



4 测试硬件:

在写驱动程序之前确信你的硬件工作正常是很重要的。使用 Wizard 程序诊断你的硬件, 如果硬件正在工作它可以查出以下内容:

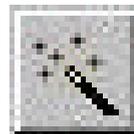
- 读写硬件的内存及寄存器。
- 监听硬件中断。



5 产生驱动程序代码:

使用 Driver Wizard 产生驱动程序代码。它可以生成如下代码:

- 操作硬件的应用程序级(内核级)API 函数。
- 用上述 API 函数操作硬件的样本应用程序。
- 各种工程文件(对应与 Windriver 所支持的各种操作系统及环境)。



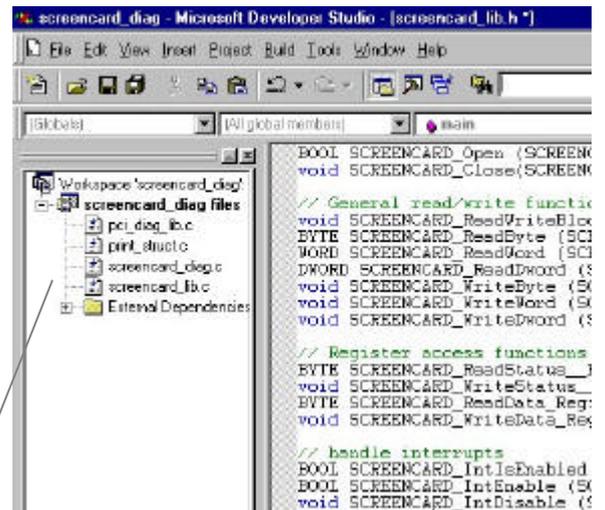
```
E MyCard_ReadStatusRegister (MyCard_HANDLE hMyCard)
```

```
TE data;  
MyCard_ReadWriteBlock(hMyCard,  
MyCard_StatusRegister_SPACE,  
MyCard_StatusRegister_OFFSET, &data,  
sizeof(BYTE), MyCard_MODE_BYTE);
```



6 编译及运行:

- 让 Driver Wizard 为你喜欢的编译器产生工程文件。
- 编译样本诊断程序并运行它! 这个样本程序是你最终的驱动程序的一个雏形。
- 修改样本程序以适合你的需要, 或者使用 WinDriver 提供的其它许多例子程序。



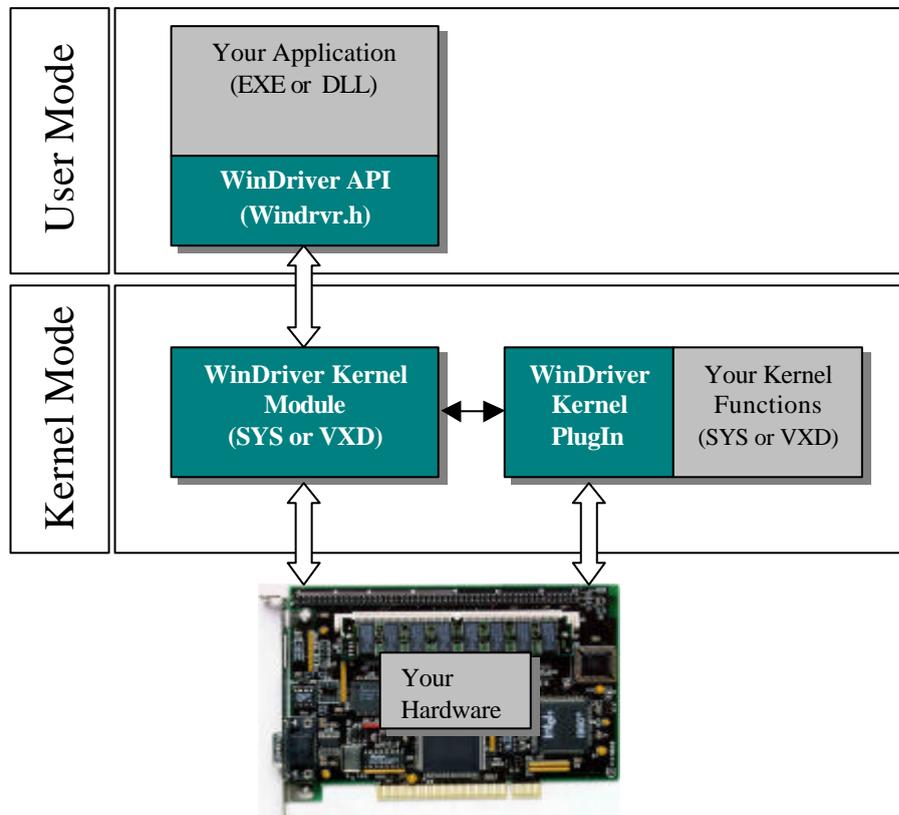
Your hardware access API,
and a sample application
that uses this API

Intuitive API. Call these functions directly from within your
application! Start with XXX_Open(), use the Read/Write
functions, Interrupt handling functions, etc., and finish with
XXX_Close().

Q: WinDriver 如何工作?

A: 使用 WinDriver, 你的设备驱动程序将在用户模式下开发(作为应用程序或一个分离的 DLL 的一部分), 你可以使用准的 Win32 开发工具(MSDEV, Borland 等等)来开发及调试你的驱动程序, 这会显著缩短开发时间。

使用 WinDriver 产生的驱动程序 (YourApp.EXE)通过 WinDriver 内核模块(Windrvr. VXD / SYS)操作你的硬件, 并使用标准的 WinDriver 函数。



Q: 使用 WinDriver 如何达到最佳性能?

A: 在驱动程序产生以后, 你可以很容易的改变驱动代码关键部分的性能 (中断或者 I/O 调用等), 方法是在 WinDriver 的核心模式下运行, 这将获得最佳性能。例如 – 在用户模式下写出中断处理过程, 写好代码, 在用户模式下调试好这段代码后, 将这段代码移到核心模式的 PlugIn 中。此时你的中断例程将在核心模式下执行, 这将达到最佳性能。

这种机制使您可以在用户模式下编写及调试代码, 然后转到核心层去运行, 代码无需任何变动。

实际操作

下面的练习将使您在 5 分钟内领略 WinDriver 的一些功能。你可以从<http://www.biaoge.com.cn> 下载一个 30 天的使用版来做如下练习。

练习 1: 读写 PCI 内存

目标: 学习如何读写 PCI 内存范围, 以及如何定义寄存器。

概况: 这个练习会演示如何通过 Driver Wizard 读写 PCI 内存卡, 并产生这样的应用程序。下面将以读写 PCI (AGP) 显示卡为例来说明这个过程。

第 1 步: 启动 Driver Wizard

第 2 步: 从开始菜单, 选择"程序 | WinDriver | Driver Wizard".

第 3 步: 在 Driver Wizard 菜单, 单击"File | New Project"菜单.

第 4 步: 从即插即用列表中选择显示卡, 通过生产商的名字来选定。

第 5 步: 单击"Memory" 标签。该显示卡三内存范围显示出来。内存范围的一段映射到显示器, 例如, 内存范围中的数据映射到屏幕上的点阵。从位置 0 开始读(屏幕的左上角位置)。现在移动一个窗口到屏幕左上角然后再读, 如果此时内存值变化了, 说明你找的内存块是正确的。现在你可以往这个内存块中写数据, 你将会发现屏幕点阵发生变化!

**** 注意:**如果你写的内存块不是显示卡位置, 将会死机并需要重启系统。

第 6 步: 定义一个"TopLeft" 寄存器, 它代表屏幕左上角 (例如位移 0)。Read and write to this register. 定义另一个寄存器 "Somewhere", 它的位移是 FF (屏幕上另一处点阵)。

第 7 步: 单击"Build | Generate code" 菜单- Driver Wizard 将产生操作硬件资源的函数, 你可以在用户模式下在你的应用程序中直接使用这些函数, 这个向导还会产生一个样本程序来使用这些函数操作硬件!

第 8 步: 编译并运行该样本程序. 使用它来读写你的显示卡。

多简单啊—试一试!

**** 注意:**你现在可以拷贝这些材料到任何其它所支持的操作系统 (Linux, Solaris, Windows CE 等等), 重新编译, 然后运行!

在练习 1 中由 Wizard 程序产生的部分 API 函数:

```
<screencard_lib.h>
```

```
BOOL SCREENCARD_Open (SCREENCARD_HANDLE *phSCREENCARD, DWORD dwVendorID, DWORD
                      dwDeviceID, DWORD nCardNum, DWORD options);
void SCREENCARD_Close(SCREENCARD_HANDLE hSCREENCARD);

// General read/write functions
void SCREENCARD_WriteWord (SCREENCARD_HANDLE hSCREENCARD, SCREENCARD_ADDR
                           addrSpace, DWORD dwOffset, WORD data);
void SCREENCARD_WriteDword (SCREENCARD_HANDLE hSCREENCARD, SCREENCARD_ADDR
                            addrSpace, DWORD dwOffset, DWORD data);

// This register represents a Pixel somewhere on the screen.
DWORD SCREENCARD_ReadSomeWhere (SCREENCARD_HANDLE hSCREENCARD);
void SCREENCARD_WriteSomeWhere (SCREENCARD_HANDLE hSCREENCARD, DWORD data);

// This register toggles the top left pixel on the screen.
DWORD SCREENCARD_ReadTopLeft (SCREENCARD_HANDLE hSCREENCARD);
void SCREENCARD_WriteTopLeft (SCREENCARD_HANDLE hSCREENCARD, DWORD data);

// handle interrupts
BOOL SCREENCARD_IntEnable (SCREENCARD_HANDLE hSCREENCARD, SCREENCARD_INT_HANDLER
                           funcIntHandler);
void SCREENCARD_IntDisable (SCREENCARD_HANDLE hSCREENCARD);
```

练习：处理中断

目标: 学习如何测试中断，并写出中断处理程序。

概况: 在这个练习中你将使用 Driver Wizard 监听由软盘驱动器产生的中断，然后你可以用此程序产生一个应用程序来监听中断并在用户模式下写出一个中断处理程序。

- 第 1: 启动 Driver Wizard
- 第 2: 从开始菜单上, 选择 "程序 | WinDriver | WinDriver Wizard".
- 第 3: 在 Wizard 的菜单上, 单击 "File | New Project". Wizard 程序会显示你机器中的即插即用卡列表
- 第 4: 由于我们将为软盘驱动器写程序, 选择 ISA
- 第 5: 单击 "中断" 标签.
 - 定义软盘驱动器: 将中断命名 "FloppyInterrupt", 选择 6 作为中断号。
 - 在创建的中断上单击右键, 单击 "shared". 这是因为要与操作系统共享此中断
- 第 6: 单击 "监听中断".
 - 查看软盘驱动器中断: 随意操作软盘驱动器 (例如, 在 DOS 路径下敲一个 "a:" 浏览 A 盘上一个文件)。
- 第 7: 单击菜单 "Build | Generate code" – Driver Wizard 将会产生你定义的中断的功能函数. 你可以在用户模式下直接调用这些函数. Wizard 还产生一个样本程序来演示如何操作软件驱动器
- 第 8: 编译并运行样本程序
- 第 9: 在样本程序中监听软件中断. 然后更改中断处理过程, 把你自己的功能加到里面去。

多简单啊—试一试!

**** 注意:**

1. 使用 WinDriver's 内核插件特性, 可以在内核模式下处理中断及 IO 调用, 这样可以达到优化操作的目的。
2. 你可以测试一下 PS 鼠标中断 (Interrupt 12 on NT)及键盘中断 (Interrupt 1 on NT).