

PCIe 设备功耗管理驱动程序的开发

王法臻, 崔少辉, 王成

(陆军工程大学石家庄校区, 河北 石家庄 050003)

摘要: 随着 PCIe 设备在计算机系统中使用数量的增多, 若能在确保设备正常工作的前提下适时调整设备的电源管理状态, 可有效降低功耗。PCIe 设备的电源管理需要依靠操作系统和驱动程序协同处理实现。通过对电源管理技术和 WDF 驱动程序框架的研究和分析, 介绍了 I/O 请求的处理流程、队列使用的原则规范等内容, 设计了基于 I/O 请求入列触发、跟随系统电源状态转换及设置框架属性的解决方案来实现设备的休眠和唤醒, 并展示部分代码和实例验证了设计的可行性。

关键词: WDF; 电源管理; 功耗控制; I/O 请求; PCIe

中图分类号: TN91; TP39

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.200046

中文引用格式: 王法臻, 崔少辉, 王成. PCIe 设备功耗管理驱动程序的开发[J]. 电子技术应用, 2020, 46(12): 89-93.

英文引用格式: Wang Fazhen, Cui Shaohui, Wang Cheng. Development of driver for PCIe device power management[J]. Application of Electronic Technique, 2020, 46(12): 89-93.

Development of driver for PCIe device power management

Wang Fazhen, Cui Shaohui, Wang Cheng

(Shijiazhang Campus of Army Engineering University, Shijiazhuang 050003, China)

Abstract: As the number of PCIe devices used in computer systems increases, timely adjusting the power management state of the device can effectively reduce power consumption. The operating system and drivers need to cooperate with each other to complete the control for the PCIe device's power management. Based on the research and analysis of the Windows operating system's power management technology and the WDF, this paper introduces the processing flow of I/O requests and the principle of queue's usage, as well as designs the solutions based on I/O request processing, power state transitions and framework property settings to achieve device hibernation and wake-up. Moreover, there are some codes shown to prove the feasibility of the design.

Key words: WDF; power management; power controlling; I/O request; PCIe

0 引言

PCIe (Peripheral Component Interconnect express) 总线能够实现与计算机系统、服务器等所有周边 I/O 设备的互连, 同前两代 I/O 总线相比, 在结构上的改变和优化使得传输速率得到了大幅度提高, 克服了 PCI 总线在系统带宽、传输速度等方面的固有缺陷, 同时具有支持热插拔、低消耗和电源管理能力等方面特点。PCIe 总线因此已得到广泛应用^[1-3]。

在计算机系统中实现对 PCIe 总线设备合理的电源管理, 及时准确地调整设备状态, 对降低功耗具有实际意义。驱动程序作为操作系统与硬件交互的中间环节, 当驱动程序具备电源管理能力时, 操作系统便能够在系统范围上管理控制设备的电源消耗。若驱动程序不具有电源管理功能, 则设备会一直处于工作状态, 也会阻止操作系统在空闲时自动进入休眠状态。

1 技术途径

1.1 驱动开发技术

使用 Windows 驱动框架 (Windows Driver Framework, WDF) 进行驱动开发是当前较为普遍的开发方式。基于对象模型提供了属性、方法和事件^[4], 同时集成封装了复杂的即插即用 (Plug and Play, PNP) 和电源管理 (Power Management, PM) 模型是 WDF 最大的特点。框架中的事物皆由对象表示, 当触发对象模型的事件后, 框架随即调用回调例程进行处理。这种面向对象、事件驱动以及默认缺省处理的设计思想, 相比上一代的 Windows 驱动模型 (Windows Driver Model, WDM) 有了极大的进步, 大大降低了开发门槛和难度^[5]。

由于框架对 PNP 和 PM 的默认配置并不能实现电源状态的调整, 且驱动程序运行在操作系统内核层中, 因此本文选择 WDF 所提供的的内核驱动程序框架 (Kernel Model Driver Framework, KMDF) 进行功耗控制驱动程序

计算机技术与应用 Computer Technology and Its Applications

的开发。采用 VS2013+WDK8.1 完成开发环境的搭建,并 利用 DebugView 软件对驱动程序 log 信息进行监视以检 查状态变化。

1.2 电源管理技术

电源管理技术规范也经历了多次调整,目前 Windows 操作系统的电源管理建立在高级配置和电源管理接口 (Advanced Configuration and Power Management Interface, ACPI)规范的基础上,可实现系统电源管理、设备电源管理、即插即用、系统管理总线控制器等功能^[6]。

系统电源状态反映了整个系统的电源消耗情况,设备 电源状态则反映了设备的电源消耗情况。根据 ACPI 规范,Windows 操作系统的电源管理支持 S0~S5 共 6 种 状态。其中,S0 为工作状态,S1~S4 为休眠状态,S5 为关 闭状态,功耗依次降低。WDF 使用与 ACPI 规范相同的 术语来描述设备电源状态,即 D0~D3 这 4 种状态。其中, D0 为工作状态,D1 和 D2 为休眠状态,D3 为关闭状态, 功耗依次降低^[7]。所有设备都应支持 D0 和 D3 状态,可 根据具体情况选择设备是否支持 D1 和 D2 状态^[8]。

系统电源状态与设备电源状态并不一定是严格的 一一对应。例如,当设备跟随系统进入工作状态时,设备 驱动程序可能会判断设备当前状态以决定是否进入低 功耗状态。

1.3 电源策略所有者

WDF 驱动框架继承了 WDM 驱动模型的层次结构, 框架中驱动程序依设备对象划分、按层分布可组成一个 设备栈,如图 1 所示。

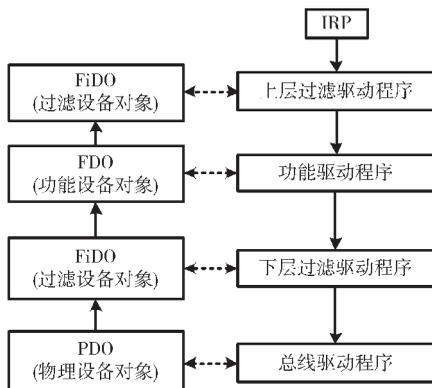


图 1 设备对象与驱动程序的层次结构

电源策略所有者 (Power Policy Owner, PPO) 是负责确 定设备电源状态,并判断电源状态需求,发送相应请求 至设备栈完成电源管理状态控制工作的驱动程序。每个 设备栈中只有一个驱动程序担任 PPO 的角色,一般是由 功能驱动程序来担当,并完成启用和禁用以下功能^[9-10]:

- (1) 当系统保持 S0 状态时,处于空闲状态的设备可 进入休眠(低功耗)状态。
- (2) 触发事件发生时,可唤醒处于休眠状态的设备以 及操作系统。

WDF 框架支持 PPO 完成以下功能:

- (1) 处理该驱动程序与设备栈其余部分之间的电源 策略通信。例如,框架将功能驱动程序更改设备电源状 态的请求传递至总线驱动程序。
- (2) 通过电源策略回调函数,启用或禁用休眠和唤醒 功能。
- (3) 提供设备属性的用户界面,启用或禁用休眠和唤 醒设置。

2 功耗控制方案的设计

功耗控制设计的核心思想就是调整设备的电源管 理状态,调整长期处于空闲状态的设备进入休眠状态, 再在需要时重新将其唤醒至工作状态。通过这种方式, 对同一类型的多个设备进行电源管理状态的切换以达 到降低功耗的目的。判断设备是否长期处于空闲状态 的关键,在于检测用户有无由操作系统向设备发送 I/O 请 求,这就需要开发人员清楚掌握 I/O 请求处理流程以及 I/O 队列的使用。

2.1 I/O 队列的使用

WDF 驱动框架隔离内核与驱动程序的特点,使得驱 动程序在一般情况下都是与框架进行通信^[11]。框架接收 到发送给设备的所有 I/O 请求都会放入 WDF 指定 I/O 队 列对象中,由队列对象统一管理并分发给驱动程序。

如图 2 所示,I/O 队列的电源管理特性会直接影响 I/O 请求的分发情况。队列是否具有电源管理的特性是 由队列配置结构体 WDF_IO_QUEUE_CONFIG 的成员变量 PowerManaged 决定的。

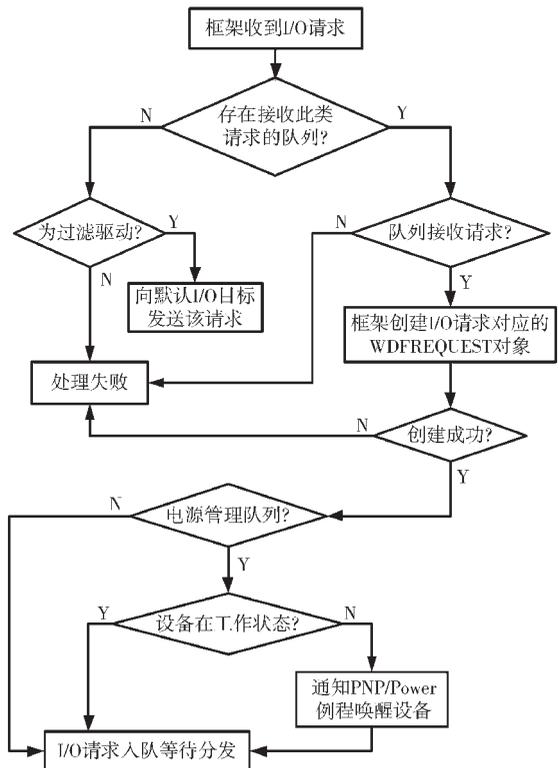


图 2 I/O 请求的处理流程

计算机技术与应用 Computer Technology and Its Applications

设置为 WdfTrue: 电源管理队列。队列对 I/O 请求的分发仅在设备处于 D0 状态时进行。若设备处于休眠状态, 则所有新进入队列的 I/O 请求都将被存储在队列中且不再向下分发。然后通知框架将设备调整至 D0 状态后再进行请求的分发。这一过程由驱动框架支持, 因而无需任何代码, 使用十分方便。

设置为 WdfFalse: 非电源管理队列。队列对 I/O 请求的分发会忽略当前设备电源状态, 直接分发请求给驱动程序, 可为无需访问设备的请求提供服务, 适用于不必与设备通信的情况, 如过滤驱动程序等。

电源管理队列的使用提供了极大的便利, 但仍需注意在驱动程序设备栈中电源管理队列的使用。如图 3 所示, 当设备处于休眠状态时, 驱动程序 1 的非电源管理队列收到系统发送的 I/O 请求可直接向下分发至驱动程序 2 的电源管理队列。由于驱动程序 2 是 PPO, 因此可直接唤醒设备并对分发请求进行处理。

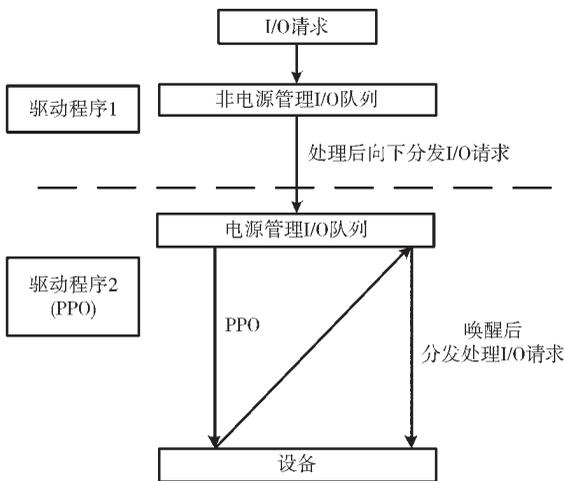


图3 上层驱动使用非电源管理队列的分析

若在 PPO 上层驱动程序 1 中强制使用电源管理队列, 会因驱动程序 1 非 PPO 而无法唤醒设备, 同时 I/O 请求无法向下分发, 导致系统在 S0 状态下向设备发送的 I/O 请求被阻塞在上层驱动程序 1 中。因此致使下层驱动程序 2(PPO) 无法接收到任何 I/O 请求, 从而无法唤醒休眠状态下的设备以恢复正常工作, 如图 4 所示。

2.2 设备功耗控制方案

功耗控制的对象是设备, 通过对设备实施电源管理以达到功耗控制的目的, 因此暂不考虑设备对系统的唤醒和空闲控制。WDF 驱动框架下的默认配置总是帮助设备设备支持 D0 和 D3 两种状态^[12]。功耗控制方案设计的焦点应集中到设备电源状态变化触发情况的分析和设置上。

KMDF 驱动框架判断设备处于空闲状态并开始计算空闲时间的依据:

(1) 电源管理队列中无排队请求, 且分发出去请求与队列无关时;

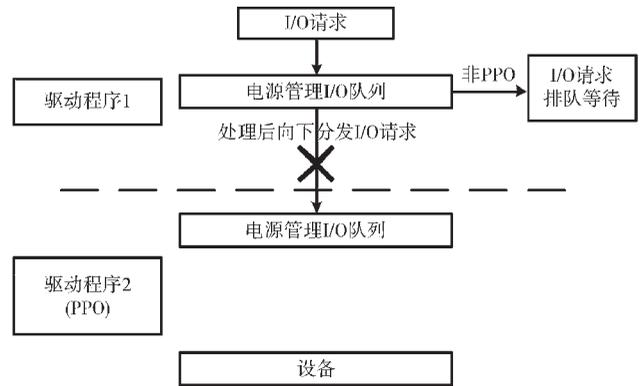


图4 上层驱动使用电源管理队列的分析

(2) 在禁用空闲功能后, 调用函数重启空闲功能时。

框架所拥有的特点使得其具备了监视 I/O 请求活动的的能力。框架可检测 I/O 活动以及队列中待处理的 I/O 对象, 当上述两种情况都不存在时, 框架便认定设备此时处于空闲状态。在使用多个 PCIE 设备的情况下, 调整空闲设备进入休眠状态, 对于降低功耗具有实际意义。

(1) 以下情况需休眠 D0 状态下的设备:

① PPO 的电源管理队列在空闲计时器规定时间内未接收到新的 I/O 请求, 同时发出的请求不再与队列有关;

② 系统从 S0 状态进入休眠状态。

(2) 以下情况需唤醒休眠状态下的设备:

① PPO 的电源管理队列接收到新的 I/O 请求;

② 用户操作设备管理器设备属性表页面禁用设备空闲功能;

③ 调用 WdfDeviceStopIdle 函数在空闲超时情况下阻止设备空闲;

④ 系统从休眠状态进入 S0 状态。

3 驱动程序功能的实现

为实现功耗控制设计方案并验证其可行性, 开发了某 PCIe 数据传输卡的驱动程序。通过 DebugView 软件查看其 log 信息以及其设备属性来验证功能的实现。

3.1 空闲功能的实现

结构体 WDF_DEVICE_POWER_POLICY_IDLE_SETTINGS 负责空闲功能的配置, 其成员变量如下:

(1) IdleCaps 表示是否支持空闲和是否在 S0 时支持唤醒, 设置为 IdleCannotWakeFromS0。

(2) DxState 表示设备空闲时电源状态, 此处设置为低功耗状态 D1 或 D2, 即 WdfPowerDeviceD1 (数值为 2) 或 WdfPowerDeviceD2 (数值为 3)。

(3) IdleTimeout 表示空闲检测时间。单位为 ms, 默认值 IdleTimeoutDefaultValue 表示 5 s。

(4) UserControlOfIdleSettings 表示是否提供设备属性页面以启用或禁用空闲功能。使用默认值 IdleAl-low-UserControl (提供属性页面)。

(5) Enabled 表示在设备空闲时是否允许其处于省电

计算机技术与应用 Computer Technology and Its Applications

状态,选择 WdfTrue 或默认值 WdfUseDefault。

(6)PowerUpIdleDeviceOnSystemWake 表示忽略 I/O 请求发送情况,始终允许设备在系统恢复至 S0 状态时恢复到 D0 状态,设置为 WdfTrue 或默认值 WdfUseDefault。这样在系统恢复 S0 状态时,设备可跟随系统完成唤醒。设备空闲功能的代码示例如下:

```
NTSTATUS PCIeEvtDeviceAdd(
_In_ WDFDRIVER Driver,
_Inout_ PWDFDEVICE DeviceInit)
{
..... //各类对象及回调例程的设置
WDF_DEVICE_POWER_POLICY_IDLE_SETTINGS
idleSettings;
WDF_DEVICE_POWER_POLICY_IDLE_SETTINGS_INIT
(&idleSettings,IdleCannotWakeFromS0);
//以空闲检测时间 50 s 为例
idleSettings.IdleTimeout=50000;
//设置设备空闲电源管理状态为 D2
idleSettings.DxState=PowerDeviceD2; WdfDeviceAssign-
S0IdleSettings(Device,&idleSettings);
.....
}
```

首先在 PCIeEvtDeviceAdd 例程中调用函数 WDF_DEVICE_POWER_POLICY_IDLE_SETTINGS_INIT 完成配置结构的初始化,设置该结构体成员变量 IdleCaps 为 IdleCannotWakeFromS0,空闲计时器检测时间为 50 s,其余变量均使用框架提供的缺省值即可。然后调用 WdfDeviceAssignS0IdleSettings 函数启用空闲功能。当框架判断设备处于空闲状态 50 s 后调整设备电源管理状态至 D2 状态。

电源例程 PCIeEvtDeviceD0Entry 和 PCIeEvtDeviceD0Exit 分别为设备进入 D0 状态和退出 D0 状态时框架所自动调用的例程。DebugView 打印信息中对这两个例程及其状态变换的监测情况如图 5 所示,当系统发起访问 PCIe 配置请求后,设备进入 D0 状态,访问结束 50 s 后退出 D0 状态。同时可通过设备属性结果验证设备状态的调整,如图 6 所示。

若需改变电源管理策略,更改结构中相应的变量值,重复上述步骤即可。

3.2 唤醒功能的实现

3.2.1 基于 I/O 请求触发框架实施电源管理

确立功能驱动程序作为电源策略所有者,I/O 队列使用电源管理队列。当系统处于 S0 状态时,新的 I/O 请求进入队列,而此时设备处于休眠状态,框架就会自动通知并完成设备电源工作状态的恢复以便处理请求。因而采用电源管理队列支持设备在系统 S0 状态下的唤醒无需编写任何代码,框架的默认设置已经提供了相应的支持以实现此功能,即当有 I/O 请求进入到队列时设备随即被唤醒。

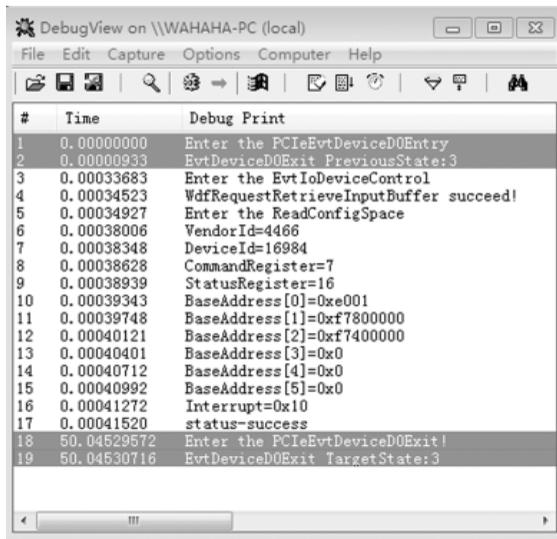


图 5 DebugView 打印信息 1



图 6 属性界面 1

如图 7 所示,在设备进入 D2 状态一段时间后,系统发起 DMA 写控制命令,设备随即由之前的 D2 状态调整至 D0 状态,而后队列分发 I/O 请求完成相关处理。同样可通过设备属性页面验证设备电源状态的调整,如图 8 所示。

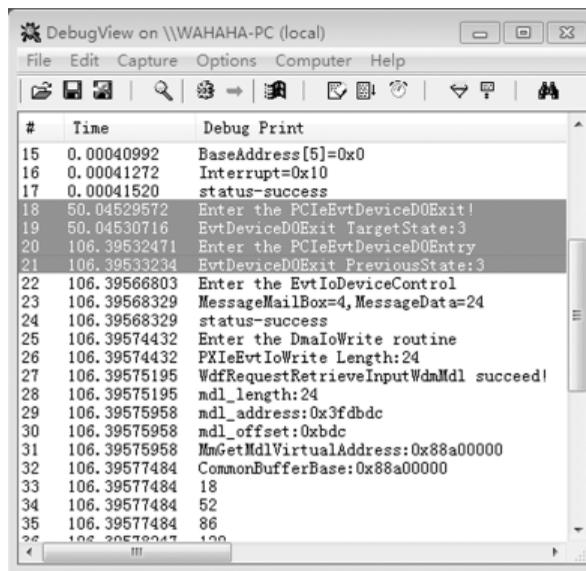


图 7 DebugView 打印信息 2



图8 属性界面2

3.2.2 禁用空闲检测功能

禁止设备空闲会将处于休眠中的设备唤醒,同时设备也将摆脱空闲检测的约束,从而使得空闲功能就此失效。

(1) 调用 WdfDeviceStopIdle 函数

在某些情况需要阻止空闲检测。例如当系统处于 S0 状态,设备句柄仍在被使用,此时不建议将设备切换至空闲状态,可通过调用函数 WdfDeviceStopIdle 来阻止设备空闲。若需要恢复空闲检测,则调用函数 WdfDeviceResumeIdle 即可重启该功能。

(2) 手动设置设备管理器属性页选项

当驱动程序允许用户界面修改空闲唤醒设置时,框架能够提供设备属性的用户界面以启用或禁用空闲和唤醒功能^[13],如图9所示。



图9 设备管理器的设备属性页面

WDF_DEVICE_POWER_POLICY_WAKE_SETTINGS 和 WDF_DEVICE_POWER_POLICY_IDLE_SETTINGS 这两个结构体可设定具有注册表访问权限的用户能够启用或禁用设备的唤醒和空闲功能。若用户界面具备控制能力,则驱动程序需将两个结构体中对应的成员变量 UserControlOfWakeSettings 和 UserControlOfIdleSettings 分别设置为 WakeAllowUserControl 和 IdleAllowUserControl,成员变量 Enabled 选择默认值即可。解决功耗控制问题暂不考虑设备唤醒系统功能,因此未使用 WDF_DEVICE_POWER_POLICY_WAKE_SETTINGS 结构体。

3.2.3 跟随系统电源状态变化实现设备的唤醒

此方法在空闲功能实现的过程中已进行相关描述,就是通过设置结构体 WDF_DEVICE_POWER_POLICY_IDLE_SETTINGS 中的成员变量 PowerUpIdleDeviceOnSystem-

Wake 来忽略 I/O 请求发送情况,始终允许设备在系统恢复至 S0 状态时恢复到 D0 状态。

4 结论

本文介绍了实现 PCIe 功耗控制所需的相关支持技术,描述了 I/O 请求在电源管理队列与非电源管理队列中的不同处理流程。在上述基础上设计了软件层控制功耗的基本解决方案,并展示了驱动程序中有关的功能代码及实现步骤,最终通过实例进行了功能的验证,在工程实践中具有一定通用性。

参考文献

- [1] 袁柳,李皓,李勤,等.基于 PCIe 高速通信接口的图像处理系统设计[J].科学技术与工程,2019,19(22):235-240.
- [2] 程鹏.基于 WDF 驱动模型的 PCIe 压缩板卡的驱动设计[D].南京:东南大学,2016.
- [3] 王帅,李念强,许惠翔.基于 Altera IP 核的 FPGA PCIe 接口设计[J].工业控制计算机,2017,30(2):41-42.
- [4] 景子龙,刘亚斌,张秀磊.基于 WDF 的 PXI 总线多功能数据采集卡驱动开发[J].电子设计工程,2017,25(11):185-188.
- [5] 樊玺玺.基于 PCI-E 总线的多功能同步数据采集卡设计[D].洛阳:河南科技大学,2015.
- [6] 王丁,王旭.台式机的低功耗之路[J].个人电脑,2015,21(6):32-43.
- [7] UEFI Forum Inc.Advanced configuration and power interface(ACPI) specification[Z].2019.
- [8] 程海全,胡君,徐抒岩,等.PCI 设备电源管理驱动程序开发[J].现代电子技术,2010(14):196-198,201.
- [9] 朱诚,左晖.利用 KMDF 驱动程序实现 USB 设备的功耗控制[J].计算机应用与软件,2012,29(12):252-254.
- [10] Microsoft.Power policy ownership[EB/OL].(2017-04-20)[2020-01-06].<https://docs.microsoft.com/zh-cn/windows-hardware/drivers/wdf/power-policy-ownership>.
- [11] 肖文君,刘万松,刘伍丰,等.基于 WDF 的 PCIe 总线驱动程序设计与实现[J].测控技术,2015,34(7):101-104.
- [12] 张小龙.基于 WDF 的 PXI-E 驱动设计[D].成都:电子科技大学,2018.
- [13] Microsoft.User control of device idle and wake behavior [EB/OL].(2017-04-20)[2020-01-16].<https://docs.microsoft.com/zh-tw/windows-hardware/drivers/wdf/user-control-of-device-idle-and-wake-behavior?redirected-from=MSDN>.

(收稿日期:2020-01-16)

作者简介:

王法臻(1992-),男,硕士研究生,主要研究方向:装备测试计量技术。

崔少辉(1965-),男,教授,主要研究方向:装备测试计量技术。

王成(1980-),男,讲师,主要研究方向:装备故障诊断技术。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所