Device Class Power Management Reference Specification

Input Device Class

V2.0

Table of Contents

Revision History

Revision	Date	Comments
0.0	7/2/96	Initial proposal for consideration.
0.5	10/24/96	Updated for internal Microsoft review.
0.9	11/6/96	Updated for 1st industry review.
0.9a	11/25/96	Incorporates feedback from 0.9 industry review plus further edits and clarifications for 2nd industry review. Taken into consideration as well are proposed Microsoft requirements and implications for USB hardware in an OnNow power-managed system, see "OnNow Power Management and USB" at http://www.microsoft.com/hwdev/pcfuture/usbdpm.htm. Note that D1 is now defined as the power conservation state for input devices (was D2 in v0.9).
0.9b	12/18/96	Incorporates feedback from 0.9a industry review, plus further edits and clarifications. Note that wake-up capability in D3 is now allowed.
0.9c	5/26/97	Further edits and clarifications, especially regarding the use of power management buttons.
0.99	7/30/97	Further edits and clarifications regarding wake-up events and i8042 keyboards.
1.0	9/5/97	Fixed errors in i8042 scan codes for power mgmt buttons; the required 0xE0 prefix was missing in all Set2 Break codes. Also, minor edit regarding advanced wake-up events.
1.0a	6/24/98	Fixed error in the i8042 set2 scan codes for the Wake key (the specified set2 scan codes did not match with how an i8042 controller would map these to set1 scan codes). **Wake event** Was previously: Set2: Make = E0, 78 Break = E0, F0, 78 **Now corrected to: Set2: Make = E0, 5E Break = E0, F0, 5E**
2.0	10/12/00	Revisions in Version 2.0 incorporate changes in power state definitions to match Appendix A of Advanced Confirmation and Power Interface Specification (ACPI) Revision 2.0.

Scope

This specification defines the behavior of the input device class as it relates to power management, and, specifically, to the four device power states defined for the OnNow Architecture. This specification applies to standard types of input devices such as keyboards, keypads, mice, pointing devices, joysticks, game pads, to devices that combine these kinds of input functionality (composite devices, etc.), and to new types of input devices such as virtual reality devices, simulation devices, etc. It is intended that input device vendors will be able to design consistent power-manageable products, and that OS vendors will be able to implement an appropriate input device power management policy based on the contents of this specification.

General Device Power Management Considerations

In the OnNow architecture, power management of individual devices is the responsibility of a policy owner in the Operating System, generally a class-specific driver. This policy-owner will implement a power conservation policy that is appropriate for devices in its class. The policy will operate in conjunction with a global system power policy implemented in the operating system (i.e. is the system Working or Sleeping?). In general, the device-class power conservation policy strives to reduce power consumption while the system is Working by transitioning amongst various available power states according to device usage.

Since the policy-owner in the Operating System has very specific knowledge of when a device is in use, or potentially in use, there is no need for hardware timers or such to determine when to make these transitions. Similarly, this level of understanding of device usage makes it possible to use fewer device power states. Generally, intermediate states attempt to draw a compromise between latency and consumption due to the uncertainty of actual device usage. With the increased knowledge in the OS, crisp decisions can be made about whether the device is needed at all. With this ability to turn devices off more frequently, the benefit of having intermediate states diminishes.

The policy-owner also determines what class-specific events can cause the system to transition from Sleeping to Working, and enables this functionality based on application or user requests. Note that the definition of the wake-up events that each class supports will influence the system's global power policy in terms of the level of power conservation the Sleeping state can attain while still meeting wake-up latency requirements set by applications or the user.

In the OnNow architecture, bus drivers also implement power policy for their bus class (e.g. PCI, USB, etc.). In general, the Bus driver has responsibility for tracking the device power states of all devices on its bus, and transitioning the Bus itself to only those power states that are consistent with those of its devices. This means that the Bus state can be no lower than the highest state of one of it's devices. However, enabled wake-up events can affect this as well. For example if a particular device is in the D2 state and set to wake-up the system, and the bus can only forward wake-up requests while in the D1 state, then the Bus must remain in the D1 state even if all devices are in a lower state.

Device power state transitions are explicitly commanded by the driver and invoked through bus-specific mechanisms (e.g. ATA Standby command, USB Suspend, etc.). In some cases, bus-specific mechanisms are not available and device-specific mechanisms must be used. Note that the explicit command for entering the D3 state may be the removal of power.

The following definitions apply to devices of all classes:

- **D0**: Device is on and running. It is receiving full power from the system, and is delivering full functionality to the user.
- **D1**: Class-specific low-power state (defined below) in which device context may or may not be lost.
- **D2**: Class-specific low-power state (defined below) in which device context may or may not be lost. Attains greater power savings than D1.
- **D3**: Device is off and not running. Device context is lost. Power may be removed from the device.

Input Device Power State Definitions

State	Status	Definition
D0	*Required*	Device is receiving full power from its power source, delivering full functionality to the user, and preserving applicable context and state

		information.
D1	*Optional*	Input device power consumption is greatly reduced. In general, device is in a power management state and is not delivering any functionality to the user except wake functionality if applicable. Device status, state, or other information indicators (for example, LEDs, LCD displays, and so on) are turned off to save power. The following device context and state information should be preserved by the policy owner or other software: **Keyboard**. Num, caps, scroll lock states (and Compose and Kana states if applicable) and associated LED/indicator states, repeat delay, and repeat rate. **Joystick**. Forced feedback effects (if applicable). **Any input device**. All context and state information that cannot be preserved by the device when it's conserving power.
D2	*N/A*	This state is not defined for input devices, use D1 as the power management state instead.
D3	*Required*	Input device is off and not running. In general, the device is not delivering any functionality to the user except wake functionality if applicable. Device context and state information is lost.

Input Device Power Conservation Policy

Present State	Next State	Cause
D3	D0	Requested by the system
D0	D1/D3*	Requested by the system (for example, system goes to sleep with wake enabled)
D0/D1	D3	Requested by the system (for example, system goes to sleep with wake disabled) Power is removed
D1/D3	D0	Device with enabled wake capability requests transition by generating a wake event Requested by the system

*Depends on capability of device (if it features D1 or D3 wake capability or not); device will be put in state with the lowest possible power consumption.

Input Device Wake-up Events

It is recommended, but not required, that input devices implement and support bus-specific wake mechanisms if these are defined for their bus type. This is recommended because a user typically uses an input device of some kind to wake the system when it is in a power management state (for example, when the system is sleeping).

The actual input data (particular button or key pressed) that's associated with a wake event should never be discarded by the device itself, but should always be passed along to the policy owner or other software for further interpretation. This software implements a policy for how this input data should be interpreted, and decides what should be passed along to higher-level software, and so on.

It is recommended that the device button(s) or key(s) used for power management purposes are clearly labeled with text and/or icons. This is recommended for keyboards and other input devices on which all buttons or keys are typically labeled with text and/or icons that identify their usage.

For example, a keyboard could include a special-purpose power management button (for example, "Power") that, when pressed during a system sleeping state, generates a wake event. Alternatively, the button(s) on mice and other pointing devices could be used to trigger a wake event.

Examples of more advanced wake events include keyboard wake signaling when any key is pressed, mouse wake signaling on detection of X/Y motion, joystick wake signaling on X/Y motion, and so on. However, in order to avoid accidental or unintentional wake of the system, and to give the user some control over which input events will result in a system wake, it's suggested that more advanced types of wake events are implemented as features that can be turned on or off by the user (for example, as part of the OSPM user interface).

Minimum Input Device Power Capabilities

An input device conforming to this specification must support the D0 and D3 states. Support for the D1 state is optional.

Recommendations for Human Interface Devices

Input devices compliant with the Human Interface Devices (HID) firmware specification v1.0 should deploy usages defined in the supplemental HID Usage Table specification v1.0 for power management buttons, i.e. POWER

and SLEEP buttons. The Generic Desktop Page (page 0x01) has been extended with the following usages compared to what's defined in the core HID 1.0 specification:

- **System Control** (Usage ID 0x80),
- System Power Down (Usage ID 0x81),
- System Sleep (Usage ID 0x82),
- System Wake Up (Usage ID 0x83).

These new usages should be used in following manner with the power management buttons:

POWER button:

- Should send usage "System Power Down" when the button is pressed to power down the system. The system power policy manager will look up the POWER button action in the current power policy (by default, shutdown) and take that action.
- Should send usage "System Wake Up" when the button is pressed again to power up the system.

SLEEP button:

- Should send usage "System Sleep" when the button is pressed to put the system to sleep. The system power policy manager will look up the SLEEP button action in the current power policy (by default, sleep) and take that action.
- Should send usage "System Wake Up" when the button is pressed again to wake up the system.

The power management buttons should be reported in a **System Control** application level collection (a.k.a. "top level" collection) in order to be interpreted correctly by the host system software. Power management buttons appearing in other application level collections will be ignored.

NOTE: In previous revisions of this specification, the "Keyboard Power" usage (index 102dec/66hex) as defined in the Keyboard/Keypad Page (page 0x07) in the core HID 1.0 specification was recommended as the way to implement a POWER button on a keyboard. However, this recommendation should not be followed anymore but has instead been replaced with the new recommendations above.

Recommendations for i8042 keyboards

i8042-based keyboards should deploy the following scan codes for power management buttons, i.e. POWER and SLEEP buttons:

Power event

Set1:	Make = E0, 5E	Break = E0, DE
Set2:	Make = E0, 37	Break = E0, F0, 37

Sleep event

Set1:	Make = E0, 5F	Break = E0, DF
Set2:	Make = E0, 3F	Break = E0, F0, 3F

Wake event

Set1:	Make = E0, 63	Break = E0, E3
Set2:	Make = E0, 5E	Break = E0, F0, 5E

The Power, Sleep, and Wake event scan codes are the i8042 equivalents to the System Power Down, System Sleep, and System Wake Up HID usages as defined above.