The Hardware Configuration Interface On Toshiba Laptops

Jonathan Buzzard `jonathan@buzzard.org.uk`

February 2000 Version 0.7

1 Overview

The Hardware Configuration Interface (HCI) is a software interface that allows applications to read or setup the hardware configurations on Toshiba portable computers. It aims to conceal differences in hardware between different models from the software.

The Hardware Configuration Interface is implemented using the System Management Mode (SMM) of the Intel processor. To call the HCI the necessary registers are loaded with the values to indicate what function we wish to ask and a value is then read from port B2h. On reading from port B2h the SMM mode of the processor is enabled, whatever function requested is carried out and execution of the software continues from immediately after the `in` instruction.

The calls to the HCI can take place at *any* time and in whatever mode the processor happens to be in at the time i.e. real, protected, virtual8086 etc. While a call to the HCI is in progress all other functions of the processor are queued pending completion of the call. This includes all maskable and *non-maskable* interrupts.

2 General

The HCI is activated by reading a byte from port B2h into the AL register. There are two functions in the HCI one that enables a device setup status to be read and another that enables it to be set. The function that is then executed depends on the value held in the AX register, according to the following list

FE00h Read current status of device
FF00h Setup device

Unlike the System Configuration Interface (SCI) it is not necessary to open or close an interface to the HCI.

3 Devices

3.1 Backlight

This function controls the backlight of the LCD panel. It enables it to be turned on and off.

Read

Call:
AX = FE00h
BX = 0002h
Return successful:
CF = 0
AH = 00h
CX = 0000h - backlight off
0001h - backlight on
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

Set

Call:
AX = FF00h
BX = 0002h
CX = 0000h - backlight off
0001h - backlight on
Return successful:
CF = 0
AH = 00h
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error
84h - Write protected

3.2 AC Adaptor

This function enables you to determine whether the laptop is being power from the AC adaptor. This information is read only.

Read

Call:
AX = FE00h
BX = 0003h
Return successful:
CF = 0
AH = 00h
CX = 0003h - AC adaptor unavailable
0004h - AC adaptor powering laptop
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

3.3 Fan

This function controls the status of the cooling fan. This method for controlling the fan works on all known models which have a fan apart from the Portage 610 and Tecra 700.

Read

Call:
AX = FE00h
BX = 0004h
Return successful:
CF = 0
AH = 00h
CX = 0000h - fan off
0001h - fan on
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

Set

Call:
AX = FF00h
BX = 0004h
CX = 0000h - fan off
0001h - fan on
Return successful:
CF = 0
AH = 00h
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error
84h - Write protected

3.4 Software suspend

The function can be used to determine if a suspend is possible i.e. is the APM BIOS enabled. It can also be used to trigger a suspend sequence exactly as in a power triggered sequence. The suspend sequence takes place regardless whether the machines power up mode is boot or resume. When powered on the next time the machine will go through a resume sequence.

Read

Call:
AX = FE00h
BX = 0010h
Return successful:
CF = 0
AH = 00h
CX = 0000h - APM disabled (suspend not possible)
0001h - APM enabled (suspend possible)
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function

Set

Call:

AX = FF00h
BX = 0010h
CX = 0001h - begin suspend sequence

Return successful:

CF = 0
AH = 00h

Return unsuccessful:

CF = 1
AH = 80h - System does not support this function
83h - Input data error
84h - Write protected

3.5 Flat Panel Information

This function is used to determine the resolution and type of the LCD flat panel installed in the system. This information is read only.

Read

Call:

AX = FE00h
BX = 0011h

Return successful:

CF = 0
AH = 00h
CH = resolution
00h - 640×480
01h - 800×600
02h - 1024×768
03h - 1024×600
04h - 800×480
CL = LCD type
00h - STN monochrome
01h - STN colour
02h - 9bit TFT
03h - 12bit TFT
04h - 18bit TFT
05h - 24bit TFT

Return unsuccessful:

CF = 1
AH = 80h - System does not support this function

3.6 SelectBay Status

This function can be used to determine what type of device if any is in the SelectBay of a laptop or docking station. The function returns the an 83h input error in under the following circumstances

- An undefined value is used in the cx register.
- On a machine that supports external SelectBay's no external SelectBay's are connected.
- On a machine that does not support external SelectBay's an attempt is made to read the status of an external SelectBay.
- On a machine that does not support $5\frac{1}{4}$" SelectBay's an attempt is made to read the status of a $5\frac{1}{4}$" SelectBay.

Read

Call:
AX = FE00h
BX = 0014h
CX = 0000h - internal SelectBay
0100h - SelectBay in docking station
0180h - $5\frac{1}{4}$" bay in docking station
Return successful:
CF = 0
AH = 00h
CX = 0000h - nothing inserted
0001h - floppy disk drive
0002h - ATAPI device such as CD-ROM or Zip
0003h - IDE device such as hard disk
0004h - 2nd battery
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

3.7 System Event FIFO

This function controls the system event FIFO. The system event is cleared after it is read and the reading pointer updated. The read function returns an 80h error if the system event function is disabled. The default state of the system event function is to be disabled.

Read

Call:
AX = FE00h
BX = 0016h
Return successful:
CF = 0
AH = 00h
CX = System event
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
8Ch - System event FIFO empty

Enable/Disable

Call:
AX = FF00h
BX = 0016h
CX = 0000h - Disable
0001h - Enable
Return successful:
CF = 0
AH = 00h
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function

3.8 Panel Status

This function reads whether the LCD panel is open or closed. It is a read-only setting.

Read

Call:
AX = FE00h
BX = 0019h
Return successful:
CF = 0
AH = 00h
CX = 0000h - Panel closed
0001h - Panel open
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function

3.9 SIR/FIR Status

This function controls whether the IrDA is in Fast (FIR) or Standard (SIR) mode.

Set

Call:
AX = FF00h
BX = 001Bh
CX = 0000h - FIR disabled/SIR enabled
0001h - FIR enabled/SIR disabled
Return successful:
CF = 0
AH = 00h
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

3.10 Display Device Status

This function determines whether the current display device is either internal, external or simultaneous.

Read

Call:
AX = FE00h
BX = 001Ch
Return successful:
CF = 0
AH = 00h
CX = 0000h - Internal
0001h - External
0002h - Simultaneous
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function

Set

Call:
AX = FF00h
BX = 001Ch
CX = 0000h - Internal
0001h - External
0002h - Simultaneous
Return successful:
CF = 0
AH = 00h
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

3.11 Hotkey event status

The function controls the Hotkey event status reporting. When enabled if a heykey is pressed (i.e. a Fn key combination) it is entered into the system event FIFO. I also believe that it causes an APM OEM extension event to be issued. Currently enabling hotkey event status reporting causes Linux to lockup if the kernel has the APM driver active. Use the System Event FIFO call (BX=0016h) to read the values in the system event FIFO.
The hotkey event value placed in the system event FIFO goes through three phases. The values placed in the system event FIFO are 16bits wide. If the Fn key has been depressed in combination with another key and the other key is *still* held down then the value placed in the FIFO will be the BIOS scan code of with bit 8 *set*. If the key has been released but the Fn key is still held down then byte will hold the BIOS scan code of the key with bit 7 and bit 8 *set*. When the Fn key has been released the value 0100h is placed into the FIFO. It should be noted that the Fn key combinations that are used to emulate keys on a standard keyboard are not entered into the system event FIFO.

Read

Call:
AX = FE00h
BX = 001Eh
Return successful:
CF = 0
AH = 00h
CX = 0000h - Disable
0001h - Enable
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function

Enable/Disable

Call:
AX = FF00h
BX = 001Eh
CX = 0000h - Disable
0001h - Enable
Return successful:
CF = 0
AH = 00h

Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

3.12 Unused Memory Information

This function sets unused memory size/address for hibernation in SM-RAM. This information can *only* be set. This function doesn't check whether the settings of the address and the size are appropriate or not. But each system checks the unused memory area number. If the unused memory area number is not supported by the system, the function returns 83h error.

Set

Call:
AX = FF00h
BX = 0021h
ECX = Physical memory address of unused area
EDX = Unused memory size
SI = Memory area number (0, 1, 2, ...)
Return successful:
CF = 0
AH = 00h
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

3.13 SelectBay Lock Status

This function enables you to determine whether the SelectBay in question is locked. This information is read only.

Read

Call:
AX = FE00h
BX = 0022h
CX = 0000h - built in device
0001h - internal SelectBay
0002h - SelectBay in docking station
0003h - $5\frac{1}{4}$" bay in docking station
Return successful:
CF = 0
AH = 00h
CX = 0000h - locked
0001h - unlocked
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

3.14 Boot Device

This function enables you to determine whether the selected device is the boot device. Ths information is read only.

Read

Call:
- AX = FE00h
- BX = 0026h
- CX = 0000h - built in device
  0001h - internal SelectBay
  0002h - SelectBay in docking station
  0003h - $5\frac{1}{4}$" bay in docking station

Return successful:
- CF = 0
- AH = 00h
- CX = 0000h - not the boot device
  0001h - boot device

Return unsuccessful:
- CF = 1
- AH = 80h - System does not support this function
  83h - Input data error

3.15 Hibernation Information

This function gets the size of the data header, maximum memory and VRAM. This enables the size of the hibernation file to be calculated. This information is read only.

Get

Call:
- AX = FE00h
- BX = 002Dh
- CX = parameter to get
  0000h - data header size (BIOS information)
  0001h - maximum memory size
  0002h - VRAM size

Return successful:
- CF = 0
- AH = 00h
- ECX = size in bytes

Return unsuccessful:
- CF = 1
- AH = 80h - System does not support this function
  83h - Input data error

3.16 Owner String

This fuction enables the owner string to be read and set. The owner string is displayed during power up. This information is tentative in that I have not yet been able to get it to work properly.

Get

Call:

AX = FE00h
BX = 0029h
SI = offset from begining of string
Return successful:
CF = 0
AH = 00h
ECX = bits 0-15 : number of valid characters
bits 16-31 : size of owner string (always 513?)
EDX = bits 0-7 : first character
bits 8-15 : second character
bits 16-23 : third character
bits 24-31 : fourth character
EDI = bits 0-7 : fifth character
bits 8-15 : sixth character
bits 16-23 : seventh character
bits 24-31 : eighth character
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function
83h - Input data error

3.17 Hibernation File Address

This gets or sets the Logical Block Address (LBA) of the hibernation data file.

Get

Call:
AX = FE00h
BX = 002Eh
Return successful:
CF = 0
AH = 00h
ECX = hibernation file address (LBA)
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function

Set

Call:
AX = FF00h
BX = 002Eh
ECX = hibernation file address (LBA)
Return successful:
CF = 0
AH = 00h
Return unsuccessful:
CF = 1
AH = 80h - System does not support this function