USING SINGLE HOTKEY TO CONTROL BRIGHTNESS ON NOTEBOOK

HP INC
Using Single Hotkey to Control Brightness on Notebook

Abstract:

- Limited Hotkey function support due to there are no enough function keys between F1 and F12 on keyboard.
- Platform designer can re-design keyboard layout to provide more hotkey space, but it causes cost up.
- Embedded controller (EC) has capability to identify the behavior when brightness hotkey is pressed or hold down within a debounce time.
- When brightness hotkey is pressed and then release within a debounce time, this hotkey event is set as brightness up.
- When brightness hotkey is kept press within a debounce time, this hotkey event is set as brightness down.
- Depend on the brightness hotkey pressed behavior that identified by EC, EC can issue scan code, SMI or ACPI Query event to BIOS to issue brightness event to ACPI OS.
- Note: This brightness hotkey debounce time can be set up in BIOS setup menu.

Design Construction:

- HW: The printing of Brightness Hotkey on Keyboard needs to re-design.
- SW: No SW need
- BIOS/EC: EC checks the brightness hotkey pressed within a debounce time and informs BIOS to notify Brightness Up/Down event to ACPI OS.
Case 1: Brightness Up - one time press
a. EC read state of brightness hotkey at stamp A and the state is "Press".
b. After the debounce time, EC read state of brightness hotkey at stamp B and state is "Release".
c. The last hotkey pressed state is "Brightness Up" and send 0x86 notification.

Case 2: Brightness Down - one time press
a. EC read state of brightness hotkey at stamp A and the state is "Press".
b. After the debounce time, EC read state of brightness hotkey at stamp B and state is "Press".
c. The last hotkey pressed state is "Brightness Down" and send 0x87 notification.

Case 3: Brightness Down - 2 times press
a. EC read state of brightness hotkey at stamp A and the state is "Press".
b. After the debounce time, EC read state of brightness hotkey at stamp B and state is "Press".
c. The first hotkey pressed state is "Brightness Down" and send 0x87 notification.
d. After the next debounce time, EC read state of brightness hotkey at stamp C and state keeps "Press".
e. The last hotkey pressed state is "Brightness Down" and send 0x87 notification.
f. Similarly, if the state of brightness hotkey still keeps "Press" in next debounce time, BIOS send 0x87 notification Continuously.
< Flow Chart >

Boot Flow

Pre-boot Environment

1. Power On
2. Get the debounce time setting of Brightness Hotkey that user defines in F30 setup menu
3. Store the debounce time in EC RAM
4. Boot to Windows OS

ACPI OS Environment

1. Press Brightness Hotkey
2. EC read the debounce time from EC RAM
3. EC read Brightness Hotkey state when counting the debounce time
4. Brightness Hotkey is released at the end of debounce time
5. Identify this hotkey event is Brightness Up
6. Store last Hotkey event into EC RAM
7. EC Issues scan code with 01 Mark code and 01 Break code

- Brightness Up
- BIOS read Hotkey Event from EC RAM
- Issue 0x86 to increase Brightness
- Issue 0x87 to decrease Brightness
- ACPI OS to Output Device
- BIOS read Hotkey Event from EC RAM
- Issue 0x86 to increase Brightness
- Issue 0x87 to decrease Brightness
Business Strategy/Advantages

1. Combine brightness up and brightness down 2 hotkeys into 1 hotkey without redesign keyboard layout.
2. Firmware solution to use single hotkey to support brightness up/down function.
3. Platform can support one more hotkey function without causing cost up.

Disclosed by Chia-Cheng Lin, Harry Chang, Matt Lin, Nung-Kai Chen, HP Inc.