

Erledigt

## DSDT Bearbeitung / Belegung der FN ACPI Tasten mit anderen Befehlen

Beitrag von „anonymous\_writer“ vom 18. Februar 2018, 17:41

Hallo DSDT Profis, 😊

Hat einer von euch eine Idee wie man die die ACPI Events anderweitig belegen kann? \_Q0A ist hier im Bild mit der Methode Rechner herunterfahren hinterlegt. Ich würde das gerne ändern auf Schlafmodus.

Wenn ich die Methode durch die in \_Q0E ersetze wird die Helligkeit mit der FN-Sleep Taste geändert. Womit ich weis das ich an der richtigen Stelle bin. Nur wie lautet die richtige Methode für Schlafmodus?

Generell wäre Interessant ob es weitere Methoden gibt die man hinter den FN Tasten hinterlegen kann. Auf dem Laptop gibt es noch Flugmodus und Autolicht welche noch frei wären.

```
Scope (_SB.PCI0.PCI-RED)
{
    Name (F000, Zero)
    Method (_Q0A, Zero)
    {
        Notify (F000, Zero)
    }
    Method (_Q0B, Zero, NotSerialized) // _Q0A EC Query
    {
        IF (GPI0 < 0x00000000)
        {
            Notify (F000, Zero)
        }
    }
    Method (_Q0C, Zero, NotSerialized) // _Q0A EC Query
    {
        IF (GPI0 < 0x00000000)
        {
            Notify (F000, Zero)
        }
    }
    Method (_Q0D, Zero, NotSerialized) // _Q0A EC Query
    {
        IF (GPI0 < 0x00000000)
        {
            Notify (F000, Zero)
        }
    }
    Method (_Q0E, Zero, NotSerialized) // _Q0A EC Query
    {
        IF (GPI0 < 0x00000000)
        {
            Notify (F000, Zero)
        }
    }
    Method (_Q0F, Zero, NotSerialized) // _Q0A EC Query
    {
        IF (GPI0 < 0x00000000)
        {
            Notify (F000, Zero)
        }
    }
    Method (_Q0G, Zero, NotSerialized) // _Q0A EC Query
    {
        IF (GPI0 < 0x00000000)
        {
            Notify (F000, Zero)
        }
    }
}
```

Beitrag von „anonymous\_writer“ vom 10. März 2018, 23:54

Habe eine Lösung gefunden und Poste die mal da diese sehr flexibel ist und auch für andere interessant sein könnte.

### Teil1:

Als erstes muß man herausbekommen in welcher Methode in der DSDT welcher ACPI Event hinterlegt ist.

Das mache ich mit diesem Kext.

<https://github.com/RehabMan/OS-X-ACPI-Debug>

Der Kext ACPIDebug.kext kopiere ich in den Clover Ordner.

Dann die DSDT mit dem Patch debug.txt und instrument\_Qxx.txt bearbeiten.

Das funktioniert natürlich nur wenn man auch \_Q Methoden in der DSDT hat. 😊

Nach einem Neustart werden einem in der Konsole die ACPI angezeigt.

### Teil2:

Hat man die Events kehrt es weiter mit diesem Kext:

<https://github.com/RehabMan/OS-X-ACPI-Keyboard>

Denn Kext ACPIKeyboard.kext starte ich bei mir aus dem Clover Ordner.

Die DSDT ( Ich nutze die ohne die Patches aus Teil 1) patchen wir mit der enthaltenen patch.txt.

In meinem Beispiel nehmen wir mal die Methode "\_Q76" welcher der FN+Flieger Taste entspricht.

So sieht diese Methode bei mir vor der Bearbeitung aus:

```
54 }
55
56 Method (_Q76, 0, NotSerialized) // _Qxx: EC Query
57 {
58     If (ATKP)
59     {
60         ^^^^ATKD.IANE (0x7A)
61     }
62 }
63
64 Method (_Q80, 0, NotSerialized) // _Qxx: EC Query
```

Und so nach der Bearbeitung:

```
1 }
2
3 Method (_Q76, 0, NotSerialized) // _Qxx: EC Query
4 {
5     Notify (RMKB, 0x2159)
6     Notify (RMKB, 0x2259)
7 }
8
9 Method (_Q80, 0, NotSerialized) // _Qxx: EC Query
```

Was habe ich gemacht? 😞

Code

1. Notify (RMKB, 0x2159)
2. Notify (RMKB, 0x2259)

hinterlegt eine Key Ausgabe "Pad 7".

Weitere Codes für andere Tasten sind:

Code

1. Notify (RMKB, 0x2155)
2. Notify (RMKB, 0x2255)

> "Pad 3"

Weitere Codes für andere Tasten sind:

Code

1. Notify (RMKB, 0x2156)
2. Notify (RMKB, 0x2256)

> "Pad 4"

Weitere Codes Weitere Codes für andere Tasten sind:

Code

1. Notify (RMKB, 0x2157)
2. Notify (RMKB, 0x2257)

> "Pad 5"

Weitere Codes Weiterer Code für andere Tasten sind:

Code

1. Notify (RMKB, 0x2158)
2. Notify (RMKB, 0x2258)

> "Pad 6"

Weiterer Code für andere Tasten sind:

Code

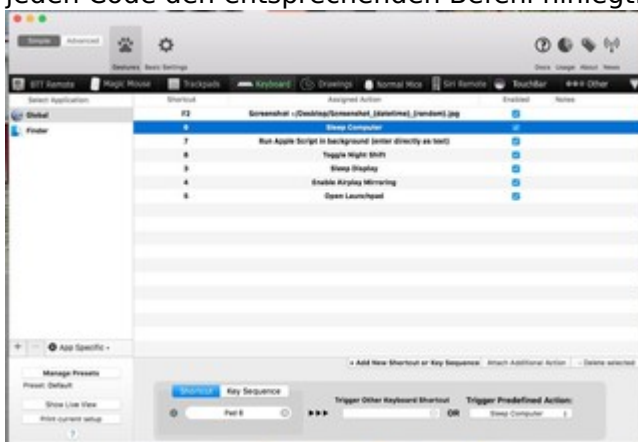
1. Notify (RMKB, 0x215B)
2. Notify (RMKB, 0x2259)

> "Pad 8"

### Teil 3:

Was mache ich jetzt mit diesen Ausgaben.

Ich liebe das Tool BetterTouchTool welches genau diese Codes versteht. So habe ich hinter jeden Code den entsprechenden Befehl hinterlegt.



Da es den Flugmodus nicht gibt habe ich dazu dieses Applescript hinterlegt. Damit wird der Status des WAN erkannt und dann entsprechend umgeschaltet.

Code

```
1. #!/usr/bin/osascript
2. #www.hackintosh-forum.de - anonymous writer
3.
4.
5. #Ermittlung Status WLAN
6. set StatusGesamt to (do shell script "networksetup -getairportpower en0")
7. set Status to last character of StatusGesamt
8.
9.
10. #WLAN Ab- oder Anschalen
11. if "n" is in Status then
12. do shell script "networksetup -setairportpower en0 off"
13. else
14. do shell script "networksetup -setairportpower en0 on"
15. end if
```

Alles anzeigen

en0 ist bei mir WLAN. Meistens ist das jedoch en1 und muß im Script dementsprechend abgeändert werden.

OK, das wars. Ist etwas mehr geworden als ich eigentlich schreiben wollte. Für mich eine coole Sache. Vielleicht möchte der ein oder andere das auch mal Testen.

<https://www.boastr.net/>

Gruß an alle die das lesen und gute Nacht. 🐼