

Neues China Hackbook Projekt: KUU K1 mit Core i5-5257U

Beitrag von „CMMChris“ vom 20. Juli 2020, 13:12

[Zitat von kuckkuck](#)

Laufen deine Brightness Keys nicht über EC Queries?

Natürlich laufen die über EC Queries. Und über den EC Query wird der entsprechende Apple Code geschickt.

Code

```
1. Scope (_SB.PCI0.LPCB.EC)
2. {
3. Method (_Q28, 0, NotSerialized) // _Qxx: EC Query, xx=0x00-0xFF
4. {
5. If (_OSI ("Darwin"))
6. {
7. Notify (\_SB.PCI0.LPCB.PS2K, 0x0406)
8. Notify (\_SB.PCI0.LPCB.PS2K, 0x10) // Reserved
9. }
10. Else
11. {
12. \_SB.PCI0.LPCB.EC.XQ28 ()
13. }
14. }
15.
16. Method (_Q29, 0, NotSerialized) // _Qxx: EC Query, xx=0x00-0xFF
17. {
18. If (_OSI ("Darwin"))
19. {
20. Notify (\_SB.PCI0.LPCB.PS2K, 0x0405)
21. Notify (\_SB.PCI0.LPCB.PS2K, 0x20) // Reserved
22. }
23. Else
24. {
25. \_SB.PCI0.LPCB.EC.XQ29 ()
26. }
```

27. }

28. }

Alles anzeigen

[Zitat von kuckkuck](#)

Hmm, wenn die Tastatur garnicht erst aktiviert wird, weil IRQs o.ä nicht ankommen, dann wird auch auf ACPI EC Query Level nichts ankommen. Also muss nicht unbedingt ein Rückschluss auf EC Firmware Probleme sein, aber kann. Hast du IRQ Fixes probiert?

Die Tastatur wird doch aktiviert, sonst würden die Brightness FN Keys doch auch nicht laufen.

IRQ Fixes habe ich nicht probiert. Hast dazu mal Lesestoff oder Vorschläge?

[Zitat von kuckkuck](#)

du kannst ja mal verschiedenes probieren und "soft" anfangen, mit einem NVRam reset, danach BIOS Reset und danach CMOS oder EC Reset.

Alles schon gemacht.

[Zitat von kuckkuck](#)

Was für ACPI Patches brauchst du für SMCBatteryManager?

Das übliche Patching der 16-Bit Werte. Über die B1B2 Methode.

Code

1. Method (B1B2, 2, NotSerialized)
2. {
3. Return ((Arg0 | (Arg1 << 0x08)))
4. }

Code

1. BDCX, 8,
2. BDCY, 8,
3. BPVX, 8,
4. BPVY, 8,
5. BRCX, 8,
6. BRCY, 8,
7. BLFX, 8,
8. BLFY, 8,
9. BSTX, 8,

10. BSTY, 8,
11. BDVX, 8,
12. BDVY, 8,
13. BACX, 8,
14. BACY, 8,

Alles anzeigen

Code

```
1. Method (_BIF, 0, NotSerialized) // _BIF: Battery Information
2. {
3.   BBIF [One] = B1B2 (BDCX, BDCY)
4.   BBIF [0x02] = B1B2 (BLFX, BLFY)
5.   BBIF [0x04] = B1B2 (BDVX, BDVY)
6.   Return (BBIF) /* \_SB_.PCI0.LPCB.H_EC.BAT0.BBIF */
7. }
```

Code

```
1. Method (_BST, 0, NotSerialized) // _BST: Battery Status
2. {
3.   Local0 = BSTS /* \_SB_.PCI0.LPCB.H_EC.BSTS */
4.   If ((B1B2 (BSTX, BSTY) & 0x10))
5.   {
6.     Local0 |= 0x04
7.   }
8.
9.   BBST [Zero] = Local0
10.  Local1 = B1B2 (BACX, BACY)
11.  If ((Local1 & 0x8000))
12.  {
13.    Local0 = (~Local1 & 0x7FFF)
14.    Local0++
15.  }
16.  Else
17.  {
18.    Local0 = (Local1 & 0x7FFF)
19.  }
20.
21.  If ((Local0 < 0x0352))
22.  {
23.    Local0 = 0x0352
24.  }
25. }
```

26. BBST [One] = Local0
27. BBST [0x02] = B1B2 (BRCX, BRCY)
28. BBST [0x03] = B1B2 (BPVX, BPVY)
29. Return (BBST) /* _SB_.PCI0.LPCB.H_EC.BAT0.BBST */
30. }

Alles anzeigen

[Zitat von kuckkuck](#)

Das wäre das erste System was ich kennenlerne, was sich mit korrekten Einstellungen nicht mit OC starten lässt.

Ich habe da stundenlang rumgetüftelt und unzählige Varianten ausprobiert. Weder mit OpenCore noch mit Clover läuft die OpenRuntime. Du kannst mir aber gerne noch ein paar Vorschläge dazu machen, da bin ich offen für Experimente. Wenn dann aber bitte erstmal mit OcQuirks für Clover. Wenn wir da Erfolg haben baue ich heute Nacht mal ne neue OC Config für das Laptop.