

OpenCore Sammelthread (lauffähige Konfigurationen Desktop - Hilfe, Diskussionen und Anregungen)

Beitrag von „Haffi86“ vom 20. Mai 2021, 10:43

[Zitat von apfelnico](#)

[Haffi86](#)

Deine Lösung greift global, alles was unter OpenCore gestartet wird - auch Windows - wird mit dieser Bearbeitung laufen müssen. Der Mainboardhersteller hat sich aber schon etwas dabei gedacht, "AWAC" als neues "Time and Alarm Device" zu priorisieren gegenüber dem alten "RTC" (AT Real-Time Clock). Das hast du aufgehoben und "RTC" darüber hinaus verändert.

Kompliziert ist das nicht, du hast ja für deinen Fall auch eine "SSDT-AWAC.aml" in die ACPI eingebunden. Es wäre lediglich eine andere SSDT notwendig. In dem Fall würde "AWAC" in Ruhe gelassen werden (stört macOS nicht) und ein für macOS benötigtes Device zusätzlich eingefügt, mittels OSI-Schleife auch nur, wenn macOS gestartet wird. Das wäre sauberer.

EDIT:

Und beides kombinieren könnte man - wenn deine ermittelten Werte für die "bad RTC region" perfekt sind - in dem man einfach nur ein in der SSDT unmodifiziertes "RTC0" einfügt. Dazu einfach das in der DSDT vorhandene Device "RTC" kopieren.

EDIT2:

Der Name "RTC0" ist einfach eine beliebige Abänderung von "RTC". Denn auch wenn "RTC" durch "AWAC" nicht aktiv ist, so ist es dennoch als Device in der DSDT vorhanden. Und da kein Device an gleicher Stelle zweimal auftauchen darf, eben diese Änderung.

Statt "SSDT-AWAC.aml" einfach eine "SSDT-RTC0" mit diesem Inhalt:

Code

```
1. External (_SB_.PCI0.LPCB, DeviceObj)
2.
3. Scope (\_SB.PCI0.LPCB)
4. {
5. Device (RTC0)
6. {
7. Name (_HID, EisaId ("PNP0B00") /* AT Real-Time Clock */) // _HID: Hardware ID
8. Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
9. {
10. IO (Decode16,
11. 0x0070, // Range Minimum
12. 0x0070, // Range Maximum
13. 0x01, // Alignment
14. 0x04, // Length
15. )
16. IO (Decode16,
17. 0x0074, // Range Minimum
18. 0x0074, // Range Maximum
19. 0x01, // Alignment
20. 0x04, // Length
21. )
22. IRQNoFlags ()
23. {8}
24. })
25. Method (_STA, 0, NotSerialized) // _STA: Status
26. {
27. If (_OSI ("Darwin"))
28. {
29. Return (0x0F)
30. }
31. Else
32. {
33. Return (Zero)
```

34. }

35. }

36. }

37. }

Alles anzeigen

wobei der Part "IO" hier nur beispielhaft ist und mit deinem Device "RTC" aus der "DSDT" abgeglichen werden müsste.

Alles anzeigen

Okay, das war sehr ausführlich. Vielen Dank dafür. Dann werde ich das mal ausprobieren.