

Erledigt

Komme bei custom DSDT-Patch (Lappy Battery) nicht weiter

Beitrag von „grt“ vom 7. Oktober 2016, 11:42

da hast du entweder einen B1B2 ... BC01 / BC02 eintrag vergessen - also den "auseinandergenommenen" ..BC.. originaleintrag dort, wo er dann verwendet/aufgerufen wird, nicht mit der B1B2 - methode versehen, oder umgekehrt: B1B2 ist vorhanden, der eintrag aber noch nicht gesplittet

nachgefragt: wenn du sowieso "zu fuss" patchst, warum dann die ganze mühe mit dem autopatch? du kannst doch die dsdt direkt editieren?

EDIT:

du hast mit dem patch weitergemacht? schieb doch mal den stand der dinge rüber. vielleicht fällt mir zum compilerfehler was ein.

noch ein EDIT:

[HIER](#) eine erkärung, wie man den batterypatch zu fuss einbauen kann (steht zwar lion dran, ist aber immer noch aktuell). für das eine 256-bit register im EC-device würde ich versuchen, zuerst 2x die methode L1L4 auf die entstandenen einzelteile anzuwenden (ergäbe 2 teile), und danach die beiden mit B1B2 wieder zu einem zusammenfügen.

könnte so aussehen:

original:

Code

1. Store (BMNX, Local5)

bearbeitet (BMNX, 256 im EC zerlegt in BM00, 8 BM1F, 8 -> "BMNX" ist nur was als beispiel ausgedachtes)

Code

1. Store (B1B2
2. ((L1L4
3. (B1B4 (BM00, BM01, BM02, BM03),
4. B1B4 (BM04, BM05, BM06, BM07),
5. B1B4 (BM08, BM09, BM0A, BM0B),
6. B1B4 (BM0C, BM0D, BM0E, BM0F)),
7. Index (Arg1, 0x0C)),
8. (L1L4
9. (B1B4 (BM10, BM11, BM12, BM13),
10. B1B4 (BM14, BM15, BM16, BM17),
11. B1B4 (BM18, BM19, BM1A, BM1B),
12. B1B4 (BM1C, BM1D, BM1E, BM1F)),
13. Index (Arg1, 0x0C))), Local5)

Alles anzeigen