

# DSDT Patches

Beitrag von „Schneelöwe“ vom 15. März 2012, 23:19

So, der nächste Teil hat jetzt aber gedauert, aber leider musste ich mich erstmal um die Grafikkarte kümmern 😊 .

Was ich in diesem Teil geplant habe ist erstmal Audio bzw. die allgemeine Device Injection per Methode \_DSM und die dazu benötigte DTGP.

Zuerst fangen wir mit dem Part der direkt das HDEF-Device betrifft, und falls ihr ein solches nicht habt, habt ihr vll. ein Gerät namens "AZAL", benennt dies einfach in HDEF um 😊 .

Also, ihr habt jetzt, je nach Karte, folgenden Code für die Karte (in meinem Fall ein sehr simpler Code für die ALC888 😞

Code

1. Method (\_DSM, 4, NotSerialized)
2. {
3. Store (Package (0x04)
4. {
5. "layout-id",
6. Buffer (0x04)
7. {
8. 0x78, 0x03, 0x00, 0x00
9. },
- 10.
- 11.
12. "PinConfigurations",
13. Buffer (Zero) {}
14. }, Local0)
15. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
16. Return (Local0)
17. }

Alles anzeigen

Doch wohin damit? Und was ist das überhaupt?

Mein gefestigtes Halbwissen sagt folgendes 😊 :

Methode (\_DSM) ist eine Methode um Daten in IOReg über die Methode DTGP "Hochzuladen".

In diesem Fall sagt der Code ungefähr folgendes: Das Gerät "HDEF", welches die Adresse 0x001B0000 (Die steht fast immer direkt unter dem Device in Name (\_ADR, xxxxxxxxxx)) hat, hat auch die layout-id von 888 und die Pinconfiguration 0. Diese Infos werden dann per DTGP

geliefert, denn wie ihr seht taucht auch DTGP dort auf.

Doch HALT! Wenn ihr aufmerksam seit ist euch vll. aufgefallen, dass ich 888 als layout-id angegeben habe, aber in `_DSM` steht doch `0x78, 0x03, 0x00, 0x00` ?

Tja, das hat primär was damit zu tun, dass 888 die id in Dezimal angegeben ist und 378 in Hex (Rechnet es doch um per [onlinetool](#) 😊 ). Doch stimmt auch die Reihenfolge nicht.

Das was in der DSDT angegeben wird, sind "Big-Endians", das was wir lesen können sind "Little-Endians".

Wie wir vom einem zu anderen kommen? Ganz einfach! Wir spiegeln die Zahl einfach in der Mitte, lassen aber die zweierpaare intakt.

Hier ein paar Beispiele:

Big-Endian: `00 02 00 00 - 14 02 00 00`

Little Endian: `00 00 02 00 - 00 00 02 14`

Ich hoffe es wurde klar 😊 .

Also, wenn wir denn Code nun einfügen, dann kriegen wir noch folgenden Fehler:

Error: Object does not exist (DTGP)

Tja, wie ich schon sagte werden die Infos aus der `_DSM`-Methode per DTGP in die IORegistry hochgeladen, aber bei uns fehlt noch die DTGP. Doch keine Panik; dafür fügen wir einfach ganz am Ende, vor der letzten geschweiften Klammer (So sieht so eine Klammer aus: `{` oder `}` ) folgendes ein:

Code

1. Method (DTGP, 5, NotSerialized)
2. {
3. If (LEqual (Arg0, Buffer (0x10)
4. {
5. /\* 0000 \*/ 0xC6, 0xB7, 0xB5, 0xA0, 0x18, 0x13, 0x1C, 0x44,
6. /\* 0008 \*/ 0xB0, 0xC9, 0xFE, 0x69, 0x5E, 0xAF, 0x94, 0x9B
7. }))
8. {
9. If (LEqual (Arg1, One))
10. {
11. If (LEqual (Arg2, Zero))
12. {
13. Store (Buffer (One)
14. {
15. 0x03
16. }, Arg4)

```
17. Return (One)
18. }
19. If (LEqual (Arg2, One))
20. {
21. Return (One)
22. }
23. }
24. }
25. Store (Buffer (One)
26. {
27. 0x00
28. }, Arg4)
29. Return (Zero)
30. }
```

Alles anzeigen

Und fertig, der Error ist weg 😊 .

Eine Anmerkung noch, die /\* \*/ sind Kommentare, alles was da zwischen steht wird ignoriert.

So nun ist erstmal genug hiermit 😊 .