

USB mittels SSDT deklarieren

Beitrag von „N0b0dy“ vom 11. Oktober 2021, 17:57

Lieber Nico oder halt [apfelnico](#) 😊

du hast hier alles gut beschrieben aber GPLD und GUPC müssen nicht entfernt werden besonders für die Leute, die gerne alle Ausgänge unter Windows oder Linux nutzen möchten, stattdessen kann man hier eine neue Methode schreiben und mittel _OSI jeden Ausgang definieren, was unter macOS machen soll.

Im Grunde genommen GUPC Methode macht nicht anders als was Name (_UPC, ...) unter macOS tut und genauso mit GPLD ist nicht anders als Name (_PLD, ...)

Hier kann man neue Methode definieren mit 2 veränderten Variablen, die so aussieht

Code

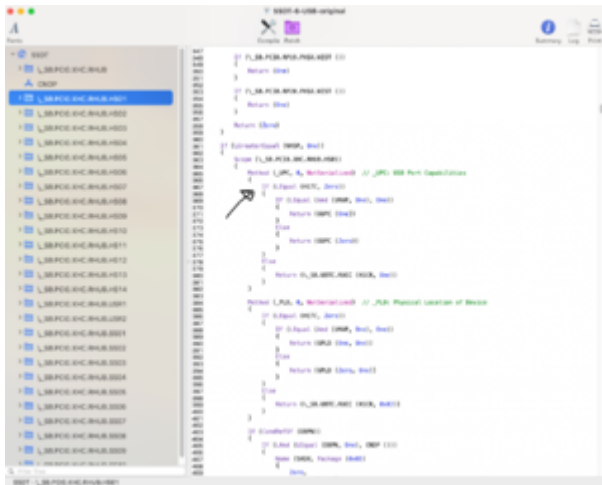
1. Method (MUPC, 2, Serialized)
2. {
3. Name (PCKG, Package (0x04)
4. {
5. Zero,
6. Zero,
7. Zero,
8. Zero
9. })
10. Store (Arg0, Index (PCKG, Zero))
11. Store (Arg1, Index (PCKG, One))
12. Return (PCKG)
13. }

Alles anzeigen

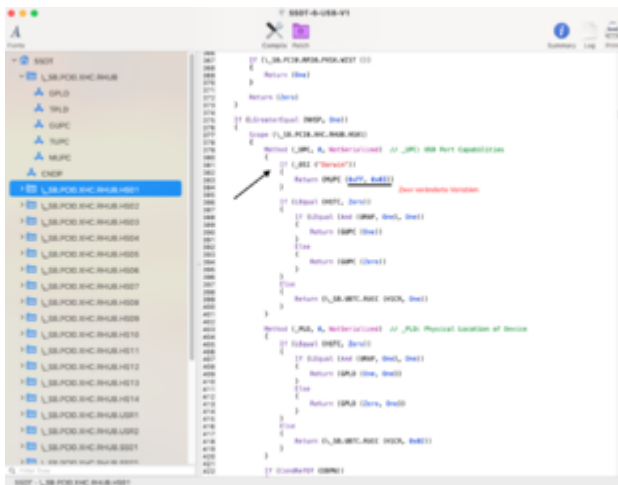
So könne wir hier der erste und zweite index von der Name (PCKG,) ändern

Danach definieren jeden Ausgang in der SSDT ohne was daran zu löschen, indem wir die Methode sagen, was sie zurückgeben soll

So sieht nicht geänderte SSdT



Und so sieht nachher



Wenn man den Ausgang unter macOS deaktivieren möchte dann einfach statt

Return (MUPC (0xff, 0x03))

nimmt

Return (MUPC (0x00, 0xff)) und so weiter....

So bleiben alle Ausgänge auch unter Windows oder Linux und nicht nur 15 Ausgänge wie unter

macOS