SSDT für USB-Ausgänge unter OpenCore ohne Kext erstellen

Beitrag von „Der_Trottel“ vom 23. Oktober 2020, 22:21

Ich versuche Anhand dieser kurze Anleitung zu zeigen, wie ihr einfach unter OpenCore eine SSDT für die USB-Ausgänge erstellt.

Was wir benötigen:

- Hackintool
- MaciASL
- 2 USB-Stick (1x nur USB2.0 und 1x USB3.0)

Erste Schritt:

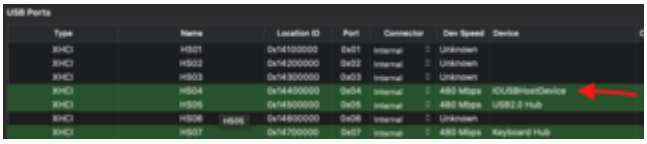
Zuerst wollen wir die USB-Ausgänge indentifizieren, welsche Nummer gehört zum welchen Ausgang, daher aktivieren wir den quirk "XhciPortLimit" Unter Kernel -> Quircks

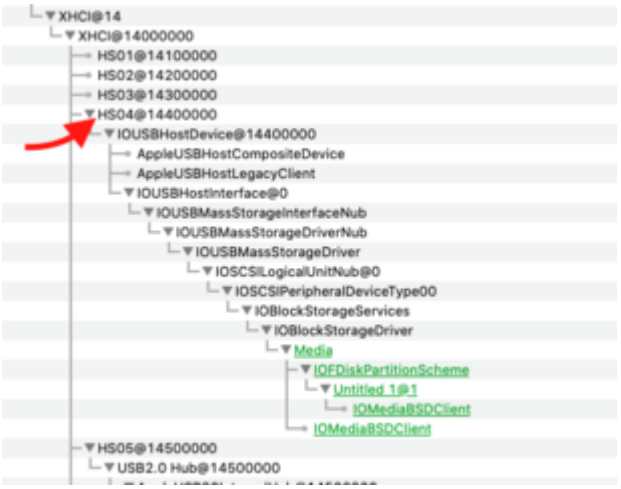
dann starten wir den PC neu.

Hier öffnen wir Hackintool und weckseln zu USB Fenster, hier werden wir alle USB-Ausgänge ansehen, wenn nicht einfach mal mit dem Tool aktualisieren. Wer mag kann auch an diese Stelle iORegistryExplorer verwenden.

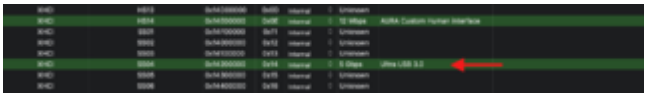
Dann stecken wir in jeden Ausgang 1x USB2.0 und USB3.0 und notieren auf einem Zettel die Nummern.

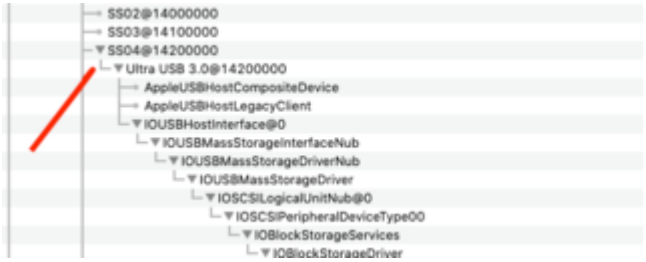
Hier an diesem Beispiel sehen wir HS04 mit USB2.0-Stick





Und hier den selben USB-Ausgang mit USB3.0 Stick





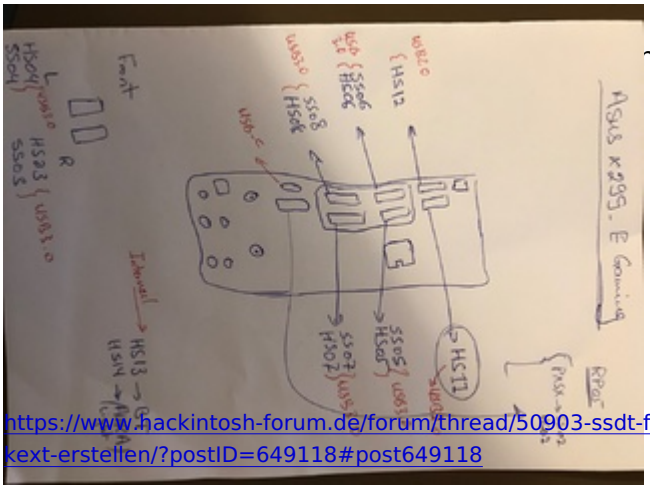
Das heißt diesen Ausgang hat die Nummer 4 und gleichzeitig besitzt USB2.0 & USB3.0, daher müssen wir diesen Ausgang nachher als USB3.0 deklarieren. Wenn dieser Ausgang nur USB2.0 hat dann müssen wir ihn als USB2.0 deklarieren

USB3.0 = 0x03

USB2.0 = 0x00

Internal = 0xFF

USB-C = 0x0A



ht

hier ist das hintere Panel für mein Asus ROG

X299-E Gaming Mainboard

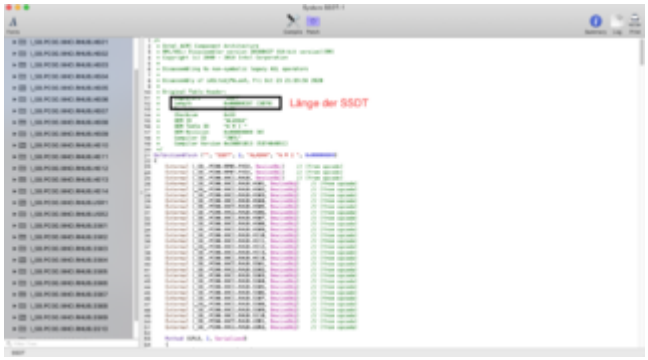
Da in macOS System die Ausgänge auf 15 begrenzt sind, dann dürfen wir nur 15 Ausgänge auswählen.

zum Beispiel habe ich mich für 3(2x), 4(2x), 5(2x), 6(2x), 7(2x), 8(2x), 11, 13, 14 entschieden, heißt es insgesamt 15.

Zweite Schritt:

Öffnen wir MaciASL und unter File -> New from ACPI sieht man alle DSDT und SSDT des Systems, hier suchen wir die SSDT für unsere USB-Ausgänge (meisten heißt es "A M I" unter X299 Mainboards und "xh_rvp" unter Zxxx Mainboards)

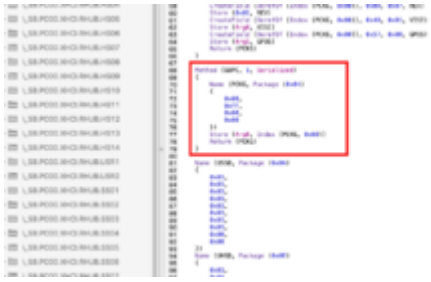
So sieht für mein Asus X299 aus



Bevor wir mit irgendwas anfangen, schreiben wir die Länge der SSDT auf, weil nachher diese SSDT in config.plist deaktivieren wollen und stattdessen unsere SSDT laden lassen.

Am Anfang steht Methode GUPC, wo jeder Ausgang es zurück gibt, daher steht für alle Ausgänge.

Hier wollen wir diese Methode für jeden Ausgang allein schreiben, deshalb löschen wir sie.



danach löschen wir die Ausgänge, die wir nicht brauchen, bei mir sieht es so aus

Unter jedem Ausgang steht die Methode "Method (_UPC, 0, NotSerialzed)", die die Methode GUPC zurück gibt, was wir vorher gesehen haben, deshalb ändern wir diese Methode zu

Code

```
1. Method (_UPC, 0, NotSerialized) // _UPC: USB Port Capabilities
2. {
3. Name (UPCP, Package (0x04)
4. {
5. One,
6. 0x03,
7. Zero,
8. Zero
9. })
10. Return (UPCP)
11. }
```

Alles anzeigen

Und zwar für jeden Ausgang, hier ist dabei zu beachten, an erste Stelle steht One und an der 2. Stelle die Codierung für deinen Ausgang, ob USB3.0(0x03) oder USB2.0(0x00) oder Internal (0xFF) ist

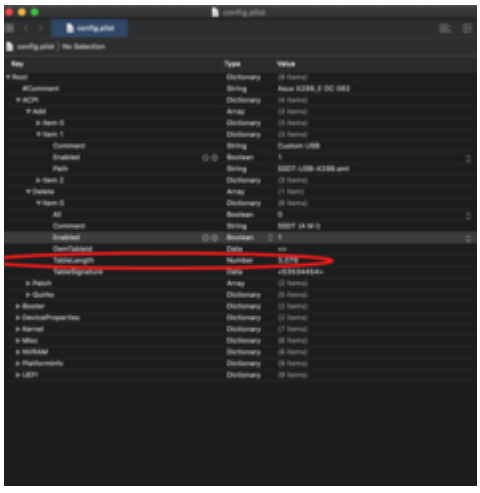
Hier ist einer Ausschnitt aus mein SSDT

danach speichern wir die SSDT als ACPI Machine Language Binary und wir geben sie irgendeine Name zum Beispiel SSDT-USB-X299.aml

Dritte Schritt:

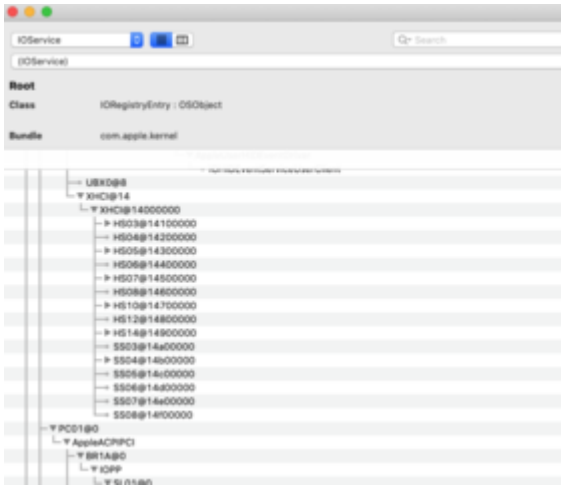
wählen wir die EFI Ordner und öffnen die config.plist, hier tragen wir unsere SSDT unter ACPI ein und SSDT selber kommt in den ACPI Ordner

Unter ACPI -> Delete deaktivieren wir die originale SSDT



danach deaktivieren wir wieder den quirk "XhciPortLimit" und starten den PC neu.

Am Ende Sieht so aus unter iOREg.


IOService
(IOService)
Root
Class
IORegistryEntry : OSObject
Bundle
com.apple.kernel
XHCI@14
XHCI@14000000
PCI1@0
AppleACPIPCI
BR1A@0
IOPP