

USBMap unter macOS

Beitrag von „N0b0dy“ vom 10. März 2023, 14:03

Allgemeine Informationen.

Seit viele Versionen ist den Quirk "XhciPortLimit" gebrochen, daher werden alle USBs nicht mehr gezeigt und ist bekannt unter macOS, dass jeden USB-Controller maximal nur 15 Ausgänge kann, deshalb muss ein USB Mapping her.

Daher ist diese kurz Anleitung ist in OpenCore wie auch Clover machbar.

Das Problem, die wir bewältigen müssen, ist fast jedes Mainboard hat insgesamt 26 Ausgänge und bei manchen Laptops noch weniger aber immer hin mehr als 15.

Deshalb müssen wir erstmal die Ausgänge identifizieren und am Ende für 15 Ausgänge entscheiden.

Jeden hinterlegten Ausgang wird mit einer Adresse gesprochen, das bedeutet wir haben 26 Adressen (hier ist die Rede vom Desktop Mainboard) und sind so geteilt

1- Hight Speed HS 1 - 14

HS01 = 0x01000000

HS02 = 0x02000000

HS03 = 0x03000000

usw..

HS14 = 0x0E000000

2- Platzhalter

USR1 = 0x0F000000

USR2 = 0x10000000

3- Super Speed SS 17 - 26

SS01 = 0x11000000

SS02 = 0x12000000

usw....

SS10 = 0x1A000000

Einfacher zu merken HS fangen mit 0 und dazu die Nummer des Ausgangs und SS fangen mit 1 und dazu die Nummer des Ausgangs, deshalb haben wir diese 2 Platzhalter.

Ziel.

Da nicht möglich alle 26 Ausgänge auf einmal zu zeigen, weil den Controller nur max. 15 zeigen kann, werden wir sie doch auf 2 teilen. Erstmal behandeln wir die High Speed (HSxx) Ausgänge danach Super Speed (SSxx). Schön daran brauchen wir kein extra externes System.

Alles, was wir brauchen:

1 - [Hackintool](#) oder [iORegistryExplorer](#)

2 - Im Anhang hinterlegte Ordner, dazu später.

3- USB-Sticks (1x USB2.0, 1x USB3.0) und wer USB-C hat dann muss es her.

Was befindet sich im Ordner?

In dem Ordner befindet sich 3 Kexts

- USBMapHSxx.kext für High Speed HSxx
- USBMapSSxx.kext für Super Speed SSxx
- USBMapAll.kext da sind alle Ausgänge hinterlegt, wo am Ende nur die USB wählen, die wir nutzen wollen

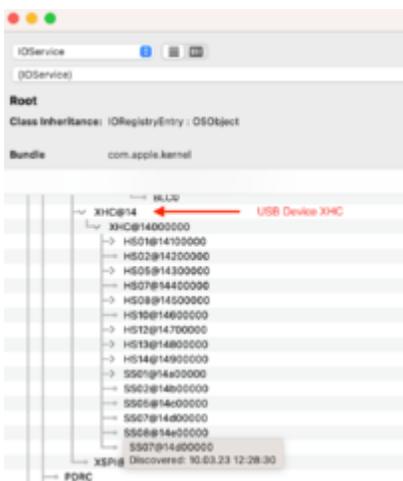
[Informationen zu Kext, die jede kennen muss, bevor wir anfangen.](#)

Hier ist ein ScreenShot von USBMapHSxx.kex



Uns ist wichtig ist den Reiter **IOKitPersonalities**, wenn wir ihn aufklappen dann sehen wir darunter hier zum Beispiel iMacPro1,1-XHC, die Name hier ist unwichtig aber besser eine Name zu wählen, worum es handelt, zum Beispiel hier handelt um SMBIOS iMacPro1,1 mit Device XHC (USB-Device)

Wenn wir den Reiter weiter aufklappen dann sehen wir weitere Zeilen, hier ist uns die Zeile mit **IONameMatch sehr wichtig** und sogar muss die gleiche Name von USB Device sein, die meisten bis Chipsatz Z490 heißen XHC, Z590 und Z690 heißen XHCI, einen Blick auf iORegistryExplorer verrät uns die Name des Devices.



Weiter hier wichtig ist die **Model** und zwar muss die gleiche Name wie SMBIOS mit dem du den Rechner fährst, auch hier wer nicht weiß welche SMBIOS hat dann kann er in EFI danach schauen

(EFI -> OC -> config.plist -> Platforminfo -> Generic -> SystemProudctName)

Dann kommen wir weiter zu **IOProviderMergeProperties**, hier haben wir **Port-count** muss gleich groß wie größte hexadezimal des USB-Ausgangs, der hier in dieser Kext hingelegt ist.



> HS12	Dictionary	4 items
> HS13	Dictionary	4 items
▼ HS14	Dictionary	4 items
comment	String	
name	String	HS14
port	Data	<0E000000> ←
UsbConnector	Number	255
model	String	IMacPro1,1

Bei **Ports** liegen alle Ausgänge, die diese Kext laden soll.

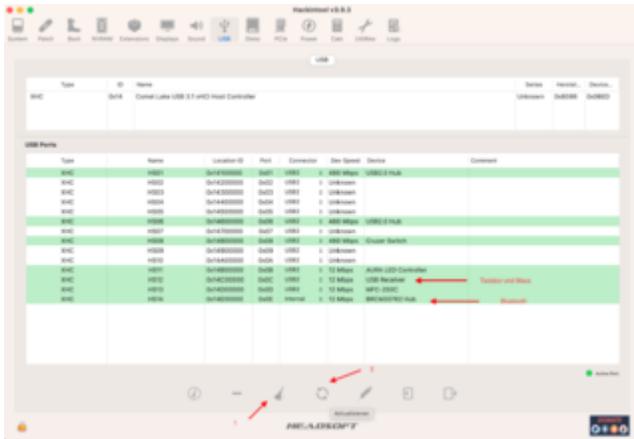
Jeder Port ist gleich aufgebaut

- **Comment** muss string sein und ist dazu info zu schreiben, die dir hilft, den Port später zu erkennen.
- **UsbConnector** muss Number sein und beschreibt den Art des Ausgangs (USB2.0 = 0, USB3.0 = 3, USB-C = 9, internal = 255)
- **Name** muss string sein und gibt die name des Ausgangs
- **Port** muss Data sein und spricht die Adresse des Ausgangs (darüber haben wir am Anfang gesprochen)

Fangen wir mal an

Wir nehmen jeglicher kext oder ssdt bezüglich USB Mapping raus und legen zu erst nur kext USBMapHSxx.kext rein und starten den Rechner neu.

Öffnen wir das Programm Hackintool oder auch iORegistryExplorer und gehen wir auf USB, hier werden wir alle HSxx sehen

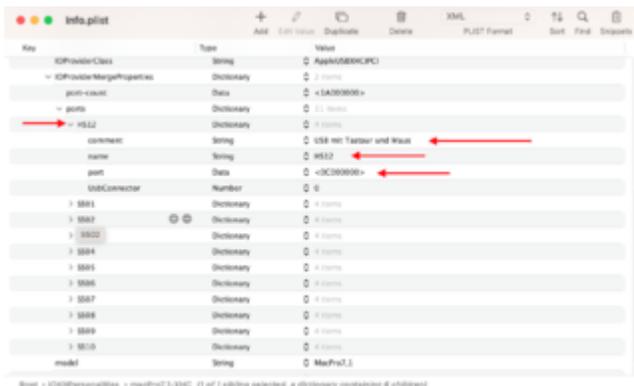


Danach nehmen wir das USB2.0 Stick und testen wir jeden Ausgang, dabei notieren wir jeden Ausgang zu welcher Nummer gehört.

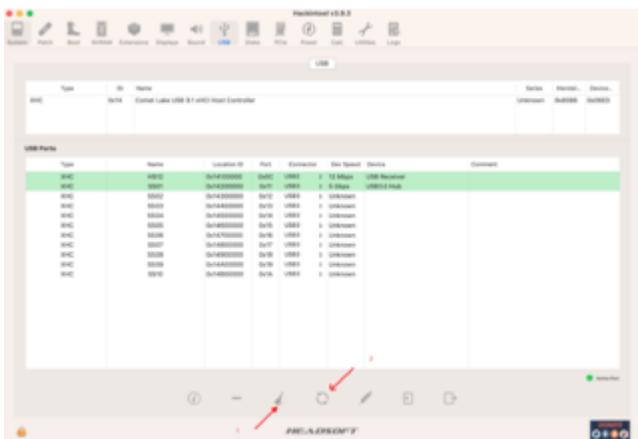
Sehr Wichtig für den nächsten Schritt: Ausgang, an dem Mouse und Tastatur befindet, brauchen wir ihn später in den USBMapSSxx.kex zu binden, sonst werden wir kein Möglichkeit haben, den Rechner nachher zu bedienen, die Leute die den Rechner mit Bluetooth bedienen, muss gleich fall sein.

Am Beispiel oben sehen wir, dass die Maus und die Tastatur am Ausgang HS12 hängen, das heißt muss diesen Ausgang in USBMapSSxx.kext enthalten. Ich habe extra in info.plist einen HS0 und HS2 hinterlegt, damit du diesen Ausgang danach auf deine passende USB für deine Tastatur und Maus ändern kannst, wer 2 Ausgänge nutzt dann muss diesen 2 Ausgänge nehmen, wer einen nutzt dann nur eins und den anderen kann gelöscht werden.

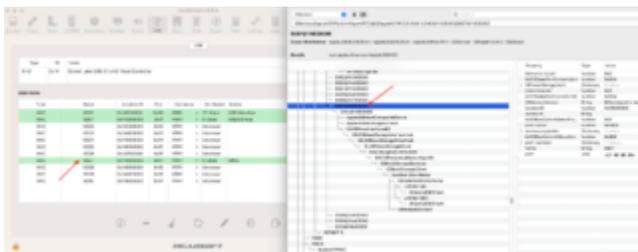
Hier ein Beispiel von diesem Beispiel



Danach nehmen wir USBMapHSxx.kext raus und binden die USBMapSSxx.ket ein und starten wir den Rechner neu, hier gleich fall wie oben, wir testen jeden Ausgang mit USB3.0 Stick und notieren wir es.



Hier ein Beispiel an einem Ausgang unter Hackintool wie auch unter iORegistryExplorer (Hier ist dir überlassen mit welcher Programm klar kommst)



Am Ende wissen wir jeden Ausgang zu welcher Nummer gehört.

Ausgänge,

die nur USB2.0 können dann deklarieren wir sie als USB2.0 also 0,

die HSxx und SSxx gleichzeitig fähig sind dann deklarieren wir sie als USB3.0 also 3,

die USB-C sind als 9

Internal wie Bluetooth als 255.

