

USB-Ports mappen unter Windows

Beitrag von „user232“ vom 13. Dezember 2021, 21:07

Ab Big Sur 11.3 müssen die USB-Ports gemappt sein. Das Programm Hackintool funktioniert ab da nicht mehr, eine Methode unter Windows möchte ich hier erläutern. (Angelehnt an den [Beitrag](#) von [LetsGo](#)). Mit dieser Methode seid ihr auch SMBIOS unabhängig.

USB-Ports mappen unter Windows

1. Download **Windows.exe** von <https://github.com/USBToolBox/tool/releases>
2. Starte dann Windows.exe
3. Drücke Taste d (Discover Ports)
4. Stecke nacheinander in jeden USB-Port einen **USB3**-Stick, immer ein bißchen warten bis er von dem Programm erkannt wird
5. Wenn alle USB-Ports durch sind drücke Taste b (Back)
6. Drücke Taste s (Select Ports and Build Kext)
7. Drücke Taste p (Enable All Populated Ports)
8. Drücke Taste k (Build UTBMap.kext)
9. UTBMap.kext wird erstellt
10. Der UTBMap.kext muss in Verbindung mit dem USBToolBox.kext <https://github.com/USBToolBox/kext/releases> (hat einen executable path und ist einfach zum downloaden!) verwendet werden. Bei Verwendung dieser Kombi sollte SSDT-RHUB auch überflüssig sein.
11. Also dann rein mit dem erstellten **UTBMap.kext** und **USBToolBox.kext** in /EFI/OC/Kexts und die config.plist wie folgt bearbeiten und dann neu starten

> Item 6	Dictionary	0 Items
> Item 7	Dictionary	0 Items
Arch	String	Any
BundlePath	String	USBToolBox.kext
Comment	String	V1.0.1
Enabled	Boolean	YES
ExecutablePath	String	Contents/MacOS/USBToolBox
MaxKernel	String	
MinKernel	String	
PlistPath	String	Contents/info.plist
> Item 8	Dictionary	0 Items
Arch	String	Any
BundlePath	String	UTBMap.kext
Comment	String	V1.1
Enabled	Boolean	YES
ExecutablePath	String	
MaxKernel	String	
MinKernel	String	
PlistPath	String	Contents/info.plist
> Item 9	Dictionary	0 Items
> Item 10	Dictionary	0 Items
> Item 11	Dictionary	0 Items

EDIT:

[LetsGo](#) hat das Thema [weiter unten](#), mal ausführlicher erläutert



Edit2:

Diese von mir erstellte Kurzanleitung lass ich mal so stehn und wird nicht mehr überarbeitet. Sofern die 15 Ports je Controller nicht überschritten werden, ist sie anwendbar. Weiteres wird ja in dem Threat diskutiert.

Beitrag von „5T33Z0“ vom 14. Dezember 2021, 00:13

[user232](#)

Punkte 10 und 11 treffen nicht zu für Windows.

USBToolbox.kext wird nur benötigt, wenn man die Ports unter macOS mapt (in Verbindung mit *UTBDefault.kex*). Am Ende entfernt man *UTBDefaul.kext* und behält *UTBMap.kext* und

USBToolobox.kext.

Siehe: <https://github.com/USBToolBox/kext#usage>

Wenn man die Ports aber unter Windows mapt und dort die Kext generiert, benötigt man am Ende nur eine Datei: UTBMap.kext

Siehe hier: <https://github.com/USBToolBox/tool#usage>

Unter Punkt 8 steht:

If building a map that uses the USBToolBox kext, make sure to grab the [latest release](#) of the kext too.

Aber da der Fall nicht zutrifft, wenn man es am PC macht, braucht man sie nicht.

Beitrag von „LetsGo“ vom 14. Dezember 2021, 01:50

5T33Z0

ist nicht ganz richtig, was du schreibst

Mit USBToolbox kann man zwei Kexte generieren (unabhängig vom Betriebssystem), die in der config.plist jedoch unterschiedlich gehandhabt werden.

1. UTBMap.kext
2. USBMap.kext

Beides sind codeless Kexte, jedoch benötigt der UTBMap.kext unbedingt noch den USBToolBox.kext (der ausführbar ist und für den man einen "executable path" in der

config.plist angeben muss)

Der USBMap.kext erweitert das von MacOS bereits bereitgestellte USB Mapping (wie hier für Original Macs unter **/System/Library/Extensions/IOUSBHostFamily.kext/Contents/PlugIns/AppleUSBHostPlatformProp** zu finden ist) entsprechend für unser System.

Die Kombi aus UTBMap.kext und USBToolBox.kext kann Vorteile (wenn man ein entsprechendes System hat) mit sich bringen, welche [hier](#) beschrieben sind:

Features

- Attach to the controller instance or parent device, allowing for more ways to match
- Ignore port definitions from ACPI to force macOS to enumerate all ports manually
 - Bypasses borked ACPI as seen on some Ryzen motherboards and 400 series Intel motherboards
 - Replaces SSDT-RHUB
- Override any built-in Apple USB maps attaching based on SMBIOS and controller name
 - Removes the need for controller renames in ACPI patches
- Does not require model identifier specified in USB map (if attaching to PCI device)
- Very compatible with existing USB maps (port format is the same)
- Does not hardcode any port maps, unlike USBInjectAll

This does **not** patch the port limit.

[user232](#)

Könntest du in deiner Beschreibung noch das Port Limit von 15 / Controller und den USBMap.kext einbauen? Vor Allem das PortLimit wäre wichtig!

Beitrag von „5T33Z0“ vom 14. Dezember 2021, 03:42

interessant. Leider geht aus den Beschreibungen der Repo nicht eindeutig her, dass man 2

Kexts mit unterschiedlichen Vorbedingungen, die zu unterschiedlichen Konfigurationen führen, generieren kann. Ebenso wenig wie, welche Extension nun alleine und welche mit 'ner extra kext verwendet werden muss...der Begriff "USBMap.kext" taucht zudem nirgends auf.

Aber zumindest wissen wir jetzt, dass es 2 mögliche Kexts mit 2 unterschiedlichen Konfigurationsmöglichkeiten und Vorbedingungen gibt und man nicht zwangsläufig die Kombination aus UTBMap.kext und USBToolBox.kext nutzen MUSS...

EDIT: [user232](#) [LetsGo](#)

Ich habe mir das Script noch mal angesehen und es ist so, dass es nicht so offensichtlich ist mit den zwei unterschiedlichen kexts:

1. Standardmäßig wird **UBTMap.kext** (USB Toolbox map kext) generiert. Diese muss mit **USBToolBox.kext** verwendet werden, wie du geschrieben hast. Diese Variante hat den Vorteil, dass diese Kombination die Ports SMBIOS und modellunabhängig mapt und in jedem macOS verwendet werden kann
2. Um aber die Variante mit **USBMap.kext** zu nutzen - die ich für die elegantere halte, da sie mit einer Kext auskommt - muss man unter erst in die Settings auf "N" (Use Native Classes) umstellen. Der Nachteil ist halt, dass diese Kext nicht SMBIOS-unabhängig ist.

Beitrag von „bluebyte“ vom 14. Dezember 2021, 08:54

Da steh' ich nun, ich armer Tor,
Und bin so klug als wie zuvor!
Heiße Magister, heiße Doktor gar,
Und ziehe schon an die zehen Jahr'
Herauf, herab und quer und krumm
Meine Schüler an der Nase herum –
Und sehe, daß wir nichts wissen können!

Quelle: Goethe, Faust. Der Tragödie erster Teil, 1808

Szene: Nacht, Faust allein in seinem gotischen Zimmer

Beitrag von „bananaskin“ vom 14. Dezember 2021, 11:36

Ich habe das auch mal ausprobiert, so wie beschrieben... komme aber nicht zum gleichen Ergebnis als unter MOJAVE mit Hackintool,

sollte man die erkannten Ports nicht auch noch deklarieren? wie in der readme des Tools beschrieben (z.B. 1,2,5,7,10 usw.)

Beitrag von „5T33Z0“ vom 14. Dezember 2021, 12:48

[Zitat von bananaskin](#)

Ich habe das auch mal ausprobiert, so wie beschrieben... komme aber nicht zum gleichen Ergebnis als unter MOJAVE mit Hackintool,

sollte man die erkannten Ports nicht auch noch deklarieren? wie in der readme des Tools beschrieben (z.B. 1,2,5,7,10 usw.)

Ausgewählt werden die Ports mit der Funktion "Enable All Populated Ports". Aber man muss dann immer noch sicherstellen, dass es nicht mehr als 15 Ports sind am Ende und ggf welche deaktivieren. Wenn der Typ des Ports falsch deklariert ist, kann man ihn mit "T" manuell anpassen, was sich mir bisher aber auch nicht ganz erschließt. Das ist in dem Guide bislang

auch noch nicht berücksichtigt worden.

Beitrag von „user232“ vom 14. Dezember 2021, 13:02

Wir sammeln hier mal alles zusammen, [LetsGo](#) schreibt eine neue Anleitung

Beitrag von „Hecatomb“ vom 14. Dezember 2021, 13:05

Geht es so also noch nicht? Ich habe vorhin einem bekannten die Anleitung geschickt und bekomme heute noch die Files für die efi. Wollten heute eigentlich noch big sur oder Monterey installieren

Beitrag von „user232“ vom 14. Dezember 2021, 13:08

Sofern die 15 Ports je Controller nicht überschritten werden, funktioniert die Anleitung. Aber wartet noch ein wenig, dann wird das Ganze auf neusten Stand gebracht.

Beitrag von „Hecatomb“ vom 14. Dezember 2021, 13:11

Naja gut... Ich sag ihm einfach er soll ein Bild von den Ports im Tool machen und lösche notfalls andere Ports aus dem kext. Danke

Beitrag von „bananaskin“ vom 14. Dezember 2021, 13:53

Hallo 5T33Z0

1. Um aber die Variante mit **USBMap.kext** zu nutzen – die ich für die elegantere halte, da sie mit einer Kext auskommt – muss man unter erst in die Settings auf "N" (Use Native Classes) umstellen. Der Nachteil ist halt, dass diese Kext nicht SMBIOS-unabhängig ist.

Wenn ich das mit OS X mache, habe ich ein Eintrag in der info.plist (siehe Bild)Setting auf "N"

Aber man sollte es doch unter Windows machen(Empfehlung) da gibt es doch kein SMBIOS

```
EQAAAA==
</data>
</dict>
<key>SS01</key>
<dict>
  <key>UsbConnector</key>
  <integer>3</integer>
  <key>port</key>
  <data>
    EQAAAA==
  </data>
</dict>
</dict>
<key>model</key>
<string>iMac19,1</string>
</dict>
</dict>
<key>OSBundleRequired</key>
<string>Root</string>
</dict>
</plist>
```

Beitrag von „LetsGo“ vom 14. Dezember 2021, 14:31

Hier mal eine Anleitung mit Bildern anhand meines Systems (Optiplex 5070 SFF)

[Hier](#) findet ihr eine exzellente allgemeine Anleitung (Information), warum wir überhaupt ein USB Mapping vornehmen müssen.

Die Bedienung funktioniert einfach mit dem jeweiligen Buchstaben gefolgt von Enter. z.B: C und Enter für das Settings Menü


```
C:\Programme (portable)\USBToolBox >
#####
#          USBToolBox 0.0.9          #
#####

Saved Data: None

D. Discover Ports
S. Select Ports and Build Kext
C. Change Settings
Q. Quit

Select an option:
```

Das Settings Menü (C):

Änderungen werden einfach mit den entsprechenden Buchstaben (T, N, A, C) und dann Enter vorgenommen

```
#####
#          Change Settings          #
#####

T. Show Friendly Types: Disabled
   Show friendly types (ie. 'USB 3 Type A') instead of numbers.
N. Use Native Classes: Enabled
   Use native Apple classes (AppleUSBHostMergeProperties) instead of the USBToolBox
   kext.
A. Add Comments to Map: Enabled
   Add port comments inside the map.
C. Bind Companions: Enabled
   Tie companion ports together. If one companion is enabled/disable/port type chang
   ed, the other companion will also be affected.
B. Back

Toggle a setting:
```

- T. Show Friendly Types: zeigt dann halt z.B: USB 3 Type A statt 3, USB 3 Type C statt 9 usw. an

```
Port 1 | USB 2.0 | USB 3 Type A (guessed)
Port 2 | USB 2.0 | Type C - with switch (guessed)
Port 3 | USB 2.0 | Type A (guessed)
- USB Keyboard - operating at USB 1.1
```

```
Port 1 | USB 2.0 | Type 3 (guessed)
Port 2 | USB 2.0 | Type 9 (guessed)
Port 3 | USB 2.0 | Type 0 (guessed)
- USB Keyboard - operating at USB 1.1
```

- N. Use Native Classes: disabled: UTBMap.kext wird erstellt, enabled: USBMap.kext wird erstellt
- A. Add Comments to Map: Man kann zu einem späteren Zeitpunkt eine Beschreibung der USB Ports hinzufügen. Bei Aktivierung wird diese Beschreibung (siehe Set Custom Names) als comment in die info.plist des erstellten Kextes übernommen.
- C. Bind Companions: Da ein physischer USB 3 Port wegen der Abwärtskompatibilität einen USB 3 und USB 2 Teil besitzt, nützt uns diese Funktion beim Mappen. Einfach USB 3 oder 2 Stick in den Port stecken und der Gegenpart wird automatisch erkannt.

Discover Ports (D):

Als Bsp. habe ich mal einen USB Stick in den Front USB 3 Port (Port 17) gesteckt um den weiteren Verlauf und Funktionen des Programms zu erklären. Dann (B)ack und in das "Select Ports and Build Kext" Menü wechseln

```
C:\Programme (portable)\USB\ x + v
#####
#                               #
#####
Intel(R) USB 3.1 eXtensible-Hostcontroller - 1.10 (Microsoft) | USB 3.0 (XHCI) | 26 ports
Port 1 | USB 2.0 | Type 3 (guessed)
Port 2 | USB 2.0 | Type 9 (guessed)
Port 3 | USB 2.0 | Type 0 (guessed)
  - USB Keyboard - operating at USB 1.1
Port 4 | USB 2.0 | Type 255 (guessed)
Port 5 | USB 2.0 | Type 0 (guessed)
  - Gaming Mouse G502 - operating at USB 1.1
Port 6 | USB 2.0 | Type 3 (guessed)
Port 7 | USB 2.0 | Type 255 (guessed)
Port 8 | USB 2.0 | Type 3 (guessed)
Port 9 | USB 2.0 | Type 0 (guessed)
Port 10 | USB 2.0 | Type 3 (guessed)
Port 11 | USB 2.0 | Type 0 (guessed)
Port 12 | USB 2.0 | Type 3 (guessed)
Port 13 | USB 2.0 | Type 255 (guessed)
Port 14 | USB 2.0 | Type 255 (guessed)
Port 15 | USB 2.0 | Type 255 (guessed)
Port 16 | USB 2.0 | Type 255 (guessed)
Port 17 | USB 3.0 | Type 3 (guessed)
  - Ultra USB 3.0 - operating at USB 3.0
Port 18 | USB 3.0 | Type 9 (guessed)
Port 19 | USB 3.0 | Type 255 (guessed)
Port 20 | USB 3.0 | Type 255 (guessed)
Port 21 | USB 3.0 | Type 3 (guessed)
Port 22 | USB 3.0 | Type 3 (guessed)
Port 23 | USB 3.0 | Type 3 (guessed)
Port 24 | USB 3.0 | Type 3 (guessed)
Port 25 | USB 3.0 | Type 255 (guessed)
Port 26 | USB 3.0 | Type 255 (guessed)

B. Back

Waiting 5 seconds:
```

Select Ports and Build Kexts (S):

Hier sieht man jetzt den praktischen Nutzen der Companion Funktion. Port 6 wurde als USB 2 Teil des Port 17 erkannt. Ich habe in diesem Fall einen USB 3 Stick angesteckt. Würde man einen USB 2 Stick verwenden, leuchtet Port 6 im "Discover Ports" Menü auf und Port 17 würde als USB 3 Teil erkannt werden.

```
C:\Programme (portable)\USBMap x + v
#####
#          Select Ports and Build Kext          #
#####
Intel(R) USB 3.1 eXtensible-Hostcontroller - 1.10 (Microsoft) | USB 3.0 (XHCI) | 4/26 ports
[ ] 1. Port 1 | USB 2.0 | Type 3 (guessed) | Companion to 22
[ ] 2. Port 2 | USB 2.0 | Type 9 (guessed) | Companion to 18
[#] 3. Port 3 | USB 2.0 | Type 0 (guessed)
    - USB Keyboard - operating at USB 1.1
[ ] 4. Port 4 | USB 2.0 | Type 255 (guessed)
[#] 5. Port 5 | USB 2.0 | Type 0 (guessed)
    - Gaming Mouse G502 - operating at USB 1.1
[#] 6. Port 6 | USB 2.0 | Type 3 (guessed) | Companion to 17
[ ] 7. Port 7 | USB 2.0 | Type 255 (guessed)
[ ] 8. Port 8 | USB 2.0 | Type 3 (guessed) | Companion to 24
[ ] 9. Port 9 | USB 2.0 | Type 0 (guessed)
[ ] 10. Port 10 | USB 2.0 | Type 3 (guessed) | Companion to 21
[ ] 11. Port 11 | USB 2.0 | Type 0 (guessed)
[ ] 12. Port 12 | USB 2.0 | Type 3 (guessed) | Companion to 23
[ ] 13. Port 13 | USB 2.0 | Type 255 (guessed)
[ ] 14. Port 14 | USB 2.0 | Type 255 (guessed)
[ ] 15. Port 15 | USB 2.0 | Type 255 (guessed)
[ ] 16. Port 16 | USB 2.0 | Type 255 (guessed)
[#] 17. Port 17 | USB 3.0 | Type 3 (guessed) | Companion to 6
    - Ultra USB 3.0 - operating at USB 3.0
[ ] 18. Port 18 | USB 3.0 | Type 9 (guessed) | Companion to 2
[ ] 19. Port 19 | USB 3.0 | Type 255 (guessed)
[ ] 20. Port 20 | USB 3.0 | Type 255 (guessed)
[ ] 21. Port 21 | USB 3.0 | Type 3 (guessed) | Companion to 10
[ ] 22. Port 22 | USB 3.0 | Type 3 (guessed) | Companion to 1
[ ] 23. Port 23 | USB 3.0 | Type 3 (guessed) | Companion to 12
[ ] 24. Port 24 | USB 3.0 | Type 3 (guessed) | Companion to 8
[ ] 25. Port 25 | USB 3.0 | Type 255 (guessed)
[ ] 26. Port 26 | USB 3.0 | Type 255 (guessed)

Binding companions is currently on.

K. Build UTBMap.kext
A. Select All
N. Select None
P. Enable All Populated Ports
D. Disable All Empty Ports
T. Show Types

B. Back

- Select ports to toggle with comma-delimited lists (eg. 1,2,3,4,5)
- Change types using this formula T:1,2,3,4,5:t where t is the type
- Set custom names using this formula C:1:Name - Name = None to clear
Select an option:
```


Die Funktionen K, A, N, P, D, T:

(K): Wäre "Use Native Classes" im Settings Menü aktiviert, würde hier "Build USBMap.kext" stehen.

(A), (N), (P), (D): eigentlich selbst erklärend

(T): Show Types: zeigt die verschiedenen Connector Typen (nähere Beschreibung samt Bildern siehe [hier!](#))

```
CA\Programme (portable)\USB
#####
# USB Types #
#####

Type A: 0
Type Mini-AB: 1
ExpressCard: 2
USB 3 Type A: 3
USB 3 Type B: 4
USB 3 Type Micro-B: 5
USB 3 Type Micro-AB: 6
USB 3 Type Power-B: 7
Type C - USB 2 only: 8
Type C - with switch: 9
Type C - without switch: 10
Internal: 255

The difference between connector types 9 and 10 is if you reverse the plug and the devices are connected to the same ports as before, they have a switch (type 9).
If not, and they are connected to different ports, they do not have a switch (type 10).

For more information and pictures, go to https://github.com/USBToolBox/tool/blob/master/TYPES.md.

B. Back

Select an option:
```

Nun zu diesen Punkten:

```
- Select ports to toggle with comma-delimited lists (eg. 1,2,3,4,5)
- Change types using this formula T:1,2,3,4,5:t where t is the type
- Set custom names using this formula C:1:Name - Name = None to clear
Select an option: t:17:0
```

-Select Ports ...: einfache einzelne Ports an und abwählen.

-Change Types: Würde das Programm z.B: einen Port falsch deklarieren, könnte man hiermit den Typ ändern. In diesem Beispiel würde t:17:0 meinen Port 17 und 6 von Typ 3 (USB 3 Typ A) in Typ 0 (USB 2 Typ A) ändern.

-Set custom names ...:

```
- Select ports to toggle with comma-delimited lists (eg. 1,2,3,4,5)
- Change types using this formula T:1,2,3,4,5:t where t is the type
- Set custom names using this formula C:1:Name - Name = None to clear
Select an option: c:17:Front USB 3
```

```
[#] 17. Port 17 | USB 3.0 | Type 3 | Companion to 6
      Front USB 3
      - Ultra USB 3.0 - operating at USB 3.0
```

mit c:17 (enter) löscht man wieder die Beschreibung

```
- Select ports to toggle with comma-delimited lists (eg. 1,2,3,4,5)
- Change types using this formula T:1,2,3,4,5:t where t is the type
- Set custom names using this formula C:1:Name - Name = None to clear
Select an option: c:17
```

Jetzt wie es z.B: anhand meines Systems aussehen sollte:

1) Discover Ports: Stick in alle benötigten Ports stecken

```
C:\Programme (portable)\USBM x + v
#####
#                               Port Discovery                               #
#####

Intel(R) USB 3.1 eXtensible-Hostcontroller – 1.10 (Microsoft) | USB 3.0 (XHCI) | 26 ports
Port 1 | USB 2.0 | Type 3 (guessed)
Port 2 | USB 2.0 | Type 9 (guessed)
Port 3 | USB 2.0 | Type 0 (guessed)
  - USB Keyboard – operating at USB 1.1
Port 4 | USB 2.0 | Type 255 (guessed)
Port 5 | USB 2.0 | Type 0 (guessed)
  - Gaming Mouse G502 – operating at USB 1.1
Port 6 | USB 2.0 | Type 3 (guessed)
Port 7 | USB 2.0 | Type 255 (guessed)
Port 8 | USB 2.0 | Type 3 (guessed)
Port 9 | USB 2.0 | Type 0 (guessed)
Port 10 | USB 2.0 | Type 3 (guessed)
Port 11 | USB 2.0 | Type 0 (guessed)
Port 12 | USB 2.0 | Type 3 (guessed)
Port 13 | USB 2.0 | Type 255 (guessed)
Port 14 | USB 2.0 | Type 255 (guessed)
Port 15 | USB 2.0 | Type 255 (guessed)
Port 16 | USB 2.0 | Type 255 (guessed)
Port 17 | USB 3.0 | Type 3 (guessed)
Port 18 | USB 3.0 | Type 9 (guessed)
Port 19 | USB 3.0 | Type 255 (guessed)
Port 20 | USB 3.0 | Type 255 (guessed)
Port 21 | USB 3.0 | Type 3 (guessed)
Port 22 | USB 3.0 | Type 3 (guessed)
Port 23 | USB 3.0 | Type 3 (guessed)
Port 24 | USB 3.0 | Type 3 (guessed)
Port 25 | USB 3.0 | Type 255 (guessed)
Port 26 | USB 3.0 | Type 255 (guessed)

B. Back

Waiting 5 seconds: |
```

2) Sieht dann so im "Select Ports ..." Menü aus:

```

Intel(R) USB 3.1 eXtensible-Hostcontroller - 1.10 (Microsoft) | USB 3.0 (XHCI) | 17/26 ports
[#] 1. Port 1 | USB 2.0 | USB 3 Type A (guessed) | Companion to 22
[#] 2. Port 2 | USB 2.0 | Type C - with switch (guessed) | Companion to 18
[#] 3. Port 3 | USB 2.0 | Type A (guessed)
    - USB Keyboard - operating at USB 1.1
    - DataTraveler 3.0 - operating at USB 2.0
    - Gaming Mouse G502 - operating at USB 1.1
[ ] 4. Port 4 | USB 2.0 | Internal (guessed)
[#] 5. Port 5 | USB 2.0 | Type A (guessed)
    - Gaming Mouse G502 - operating at USB 1.1
    - DataTraveler 3.0 - operating at USB 2.0
    - USB Keyboard - operating at USB 1.1
[#] 6. Port 6 | USB 2.0 | USB 3 Type A (guessed) | Companion to 17
[ ] 7. Port 7 | USB 2.0 | Internal (guessed)
[#] 8. Port 8 | USB 2.0 | USB 3 Type A (guessed) | Companion to 24
    - DataTraveler 3.0 - operating at USB 2.0
[#] 9. Port 9 | USB 2.0 | Type A (guessed)
    - Ultra USB 3.0 - operating at USB 2.0
[#] 10. Port 10 | USB 2.0 | USB 3 Type A (guessed) | Companion to 21
    rechts hinten über Ethernet
    - DataTraveler 3.0 - operating at USB 2.0
[#] 11. Port 11 | USB 2.0 | Type A (guessed)
    - Ultra USB 3.0 - operating at USB 2.0
[#] 12. Port 12 | USB 2.0 | USB 3 Type A (guessed) | Companion to 23
[ ] 13. Port 13 | USB 2.0 | Internal (guessed)
[#] 14. Port 14 | USB 2.0 | Internal (guessed)
    - BRCM20702 Hub - operating at USB 1.1
    - Bluetooth USB Host Controller - operating at USB 1.1
[ ] 15. Port 15 | USB 2.0 | Internal (guessed)
[ ] 16. Port 16 | USB 2.0 | Internal (guessed)
[#] 17. Port 17 | USB 3.0 | USB 3 Type A (guessed) | Companion to 6
    - Ultra USB 3.0 - operating at USB 3.0
[#] 18. Port 18 | USB 3.0 | Type C - with switch (guessed) | Companion to 2
    - Ultra USB 3.0 - operating at USB 3.0
[ ] 19. Port 19 | USB 3.0 | Internal (guessed)
[ ] 20. Port 20 | USB 3.0 | Internal (guessed)
[#] 21. Port 21 | USB 3.0 | USB 3 Type A (guessed) | Companion to 10
    rechts hinten über Ethernet
    - DataTraveler 3.0 - operating at USB 3.0
[#] 22. Port 22 | USB 3.0 | USB 3 Type A (guessed) | Companion to 1
    - DataTraveler 3.0 - operating at USB 3.0
[#] 23. Port 23 | USB 3.0 | USB 3 Type A (guessed) | Companion to 12
    - DataTraveler 3.0 - operating at USB 3.0
[#] 24. Port 24 | USB 3.0 | USB 3 Type A (guessed) | Companion to 8
    - DataTraveler 3.0 - operating at USB 3.0
[ ] 25. Port 25 | USB 3.0 | Internal (guessed)
[ ] 26. Port 26 | USB 3.0 | Internal (guessed)

Binding companions is currently off.

```

Für eine Broadcom Karte (in meinem Fall BCM94360NG) muss das Ganze so aussehen.


```
[#] 14. Port 14 | USB 2.0 | Internal (guessed)
    - BCM20702 Hub - operating at USB 1.1
    - Bluetooth USB Host Controller - operating at USB 1.1
```

3) Um das Portlimit von 15 einzuhalten deaktiviere ich die Companion Funktion im Settings Menü und wähle den USB 2 Companion Port (Port 2) vom Front USB-C Port und Port 21 (da hängt eh nur Peripherie mit USB 2 Speed dran) ab. Dazu einfach 2,21 eingeben und Enter drücken (wäre Bind Companions aktiviert, würde damit Port 2 und 18 sowie 23 und 10 deaktiviert werden)

```
- Select ports to toggle with comma-delimited lists (eg. 1,2,3,4,5)
- Change types using this formula T:1,2,3,4,5:t where t is the type
- Set custom names using this formula C:1:Name - Name = None to clear
Select an option: 2,21
```

```
C:\Programme (portable)\US  X  +  v
#####
#          Select Ports and Build Kext          #
#####

Intel(R) USB 3.1 eXtensible-Hostcontroller - 1.10 (Microsoft) | USB 3.0 (XHCI) | 15/26 ports
[#] 1. Port 1 | USB 2.0 | USB 3 Type A (guessed) | Companion to 22
[ ] 2. Port 2 | USB 2.0 | Type C - with switch (guessed) | Companion to 18
[#] 3. Port 3 | USB 2.0 | Type A (guessed)
    - USB Keyboard - operating at USB 1.1
    - DataTraveler 3.0 - operating at USB 2.0
    - Gaming Mouse G502 - operating at USB 1.1
[ ] 4. Port 4 | USB 2.0 | Internal (guessed)
[#] 5. Port 5 | USB 2.0 | Type A (guessed)
    - Gaming Mouse G502 - operating at USB 1.1
    - DataTraveler 3.0 - operating at USB 2.0
    - USB Keyboard - operating at USB 1.1
[#] 6. Port 6 | USB 2.0 | USB 3 Type A (guessed) | Companion to 17
[ ] 7. Port 7 | USB 2.0 | Internal (guessed)
[#] 8. Port 8 | USB 2.0 | USB 3 Type A (guessed) | Companion to 24
    - DataTraveler 3.0 - operating at USB 2.0
[#] 9. Port 9 | USB 2.0 | Type A (guessed)
    - Ultra USB 3.0 - operating at USB 2.0
[#] 10. Port 10 | USB 2.0 | USB 3 Type A (guessed) | Companion to 21
    rechts hinten über Ethernet
    - DataTraveler 3.0 - operating at USB 2.0
[#] 11. Port 11 | USB 2.0 | Type A (guessed)
    - Ultra USB 3.0 - operating at USB 2.0
[#] 12. Port 12 | USB 2.0 | USB 3 Type A (guessed) | Companion to 23
[ ] 13. Port 13 | USB 2.0 | Internal (guessed)
[#] 14. Port 14 | USB 2.0 | Internal (guessed)
    - BRCM20702 Hub - operating at USB 1.1
    - Bluetooth USB Host Controller - operating at USB 1.1
[ ] 15. Port 15 | USB 2.0 | Internal (guessed)
[ ] 16. Port 16 | USB 2.0 | Internal (guessed)
[#] 17. Port 17 | USB 3.0 | USB 3 Type A (guessed) | Companion to 6
    - Ultra USB 3.0 - operating at USB 3.0
[#] 18. Port 18 | USB 3.0 | Type C - with switch (guessed) | Companion to 2
    - Ultra USB 3.0 - operating at USB 3.0
[ ] 19. Port 19 | USB 3.0 | Internal (guessed)
[ ] 20. Port 20 | USB 3.0 | Internal (guessed)
[ ] 21. Port 21 | USB 3.0 | USB 3 Type A (guessed) | Companion to 10
    rechts hinten über Ethernet
    - DataTraveler 3.0 - operating at USB 3.0
[#] 22. Port 22 | USB 3.0 | USB 3 Type A (guessed) | Companion to 1
    - DataTraveler 3.0 - operating at USB 3.0
[#] 23. Port 23 | USB 3.0 | USB 3 Type A (guessed) | Companion to 12
    - DataTraveler 3.0 - operating at USB 3.0
[#] 24. Port 24 | USB 3.0 | USB 3 Type A (guessed) | Companion to 8
    - DataTraveler 3.0 - operating at USB 3.0
[ ] 25. Port 25 | USB 3.0 | Internal (guessed)
[ ] 26. Port 26 | USB 3.0 | Internal (guessed)
```

4) Nun erstelle ich beide Kexte (UTBMap.kext und USBMap.kext) mit (K) Enter. Einfach im Settings Menü (Use Native Classes)umstellen. Bei Erstellung des USBMap.kext wird noch nach dem SMBIOS gefragt: ich nutze z.B: iMacPro1,1

```

C:\Programme (portable)\USB
#####
#           Enter Model Identifier           #
#####
You are seeing this as you have selected to use AppleUSBHostController. Model identifier autodetection is unavailable as you are not on macOS.
Please enter the model identifier of the target system below. You can find it in System Information or with 'system_profiler -detailLevel mini SPHardwareDataType'.
Enter the model identifier: iMacPro1,1

```

5) Somit erhält man die generierten Kexte im USBToolBox Verzeichnis:

USBMap.kext	14.12.2021 12:37	Dateiordner	
UTBMap.kext	14.12.2021 12:36	Dateiordner	
settings.json	14.12.2021 12:36	JSON-Datei	1 KB
usb.json	14.12.2021 12:37	JSON-Datei	21 KB
Windows.exe	12.06.2021 14:41	Anwendung	11 591 KB

6) benutzt man UTBMap.kext muss man noch zusätzlich den USBToolBox.kext von [Github](https://github.com) runterladen und die beiden in den Kexts Ordner der EFI kopieren und dann in die config.plist einbinden. Dabei aufpassen, dass der USBToolBox.kext einen ausführbaren Pfad hat.

<ul style="list-style-type: none"> ▼ 11 BundlePath Comment Enabled ExecutablePath Arch MaxKernel MinKernel PlistPath 	<ul style="list-style-type: none"> ‡ Dictionary ‡ String ‡ String ‡ Boolean ‡ String ‡ String ‡ String ‡ String ‡ String 	<ul style="list-style-type: none"> 8 key/value pairs USBToolBox.kext USBToolBox.kext YES Contents/MacOS/USBToolBox Any Contents/Info.plist
<ul style="list-style-type: none"> ▼ 12 BundlePath Comment Enabled ExecutablePath Arch MaxKernel MinKernel PlistPath 	<ul style="list-style-type: none"> ‡ Dictionary ‡ String ‡ String ‡ Boolean ‡ String ‡ String ‡ String ‡ String ‡ String 	<ul style="list-style-type: none"> 8 key/value pairs UTBMap.kext UTBMap.kext YES Any Contents/Info.plist

Für den USBMap.kext wird USBToolbox.kext nicht benötigt, da dieser die nativen Apple Kexte benutzt.

Zum Schluss sei angemerkt, dass man sich einfach ein wenig mit dem Programm vertraut machen sollte. Keine Angst, man kann nichts kaputt machen 😊

Beitrag von „bananaskin“ vom 14. Dezember 2021, 14:48

Hey Super....damit sollten wir klar kommen, sehr gut erklärt 🤖👍👍



Beitrag von „5T33Z0“ vom 14. Dezember 2021, 15:58

[Zitat von bananaskin](#)

Hallo 5T33Z0

1. Um aber die Variante mit **USBMap.kext** zu nutzen – die ich für die elegantere halte, da sie mit einer Kext auskommt – muss man unter erst in die Settings auf "N" (Use Native Classes) umstellen. Der Nachteil ist halt, dass diese Kext nicht SMBIOS-unabhängig ist.

Wenn ich das mit OS X mache, habe ich ein Eintrag in der info.plist (siehe Bild)Setting auf "N"

Aber man sollte es doch unter Windows machen(Empfehlung) da gibt es doch kein SMBIOS

```
    </data>
  </dict>
  <key>SS01</key>
  <dict>
    <key>UsbConnector</key>
    <integer>3</integer>
    <key>port</key>
    <data>
      EQAAAA==
    </data>
  </dict>
</dict>
</dict>
<key>model</key>
<string>iMac19,1</string>
</dict>
<key>OSBundleRequired</key>
<string>Root</string>
</dict>
</plist>
```

Ähhh, deine Argumentation ist redundant. Es ist vollkommen egal, ob man das unter Windows oder macOS macht. Windows verwendet weder SMBIOS noch Kexts und **dennoch** kann es Kexts mit Configurations-Dateien generieren und da was weiß ich reinschreiben. Das Tool basiert doch auf irgend 'ner scriptsprache die einfach generiert, was man ihm sagt - Betriebssystem-unabhängig - Empfehlung hin oder her.

Beitrag von „plutect“ vom 14. Dezember 2021, 18:56

Ist es besser wenn man Windows zum Mappen per F12 direkt startet oder mit OpenCore, egal ?

```
#####
#                               #
#####
Intel(R) 7 Series/C230-Chipset/familie - USB-erweiterter Hostcontroller - 1826 | USB 2.0 (EHCI) | 2 ports
Port 1 | USB 2.0 | Internal (guessed)
  - Generic USB Hub - operating at USB 2.0
  - Dell Multimedia Pro Keyboard Hub - operating at USB 1.1
  - Dell Multimedia Pro Keyboard - operating at USB 1.1
  - Dell USB Optical Mouse - operating at USB 1.1
Port 2 | USB 2.0 | Type A (guessed)
Intel(R) USB 3.0 extensible-Hostcontroller - 1.0 (XHCI) | 8 ports
Port 1 | USB 2.0 | USB 3 Type A (guessed)
Port 2 | USB 2.0 | USB 3 Type A (guessed)
Port 3 | USB 2.0 | USB 3 Type A (guessed)
Port 4 | USB 2.0 | USB 3 Type A (guessed)
Port 5 | USB 3.0 | USB 3 Type A (guessed)
Port 6 | USB 3.0 | USB 3 Type A (guessed)
Port 7 | USB 3.0 | USB 3 Type A (guessed)
Port 8 | USB 3.0 | USB 3 Type A (guessed)
Intel(R) 7 Series/C230-Chipset/familie - USB-erweiterter Hostcontroller - 1820 | USB 2.0 (EHCI) | 2 ports
Port 1 | USB 2.0 | Internal (guessed)
  - Generic USB Hub - operating at USB 2.0
  - Bluetooth USB Host Controller - operating at USB 1.1
Port 2 | USB 2.0 | Type A (guessed)
0. Back
waiting 5 seconds:
```

Bei Port 2 von EHCI weiß ich schon nicht weiter, dieser zeigt keine Geräte an das ist ein USB 2.0 A Port der intern an meinem QUO Board an USB1 angeschlossen ist, wenn ich den aber rausnehme geht der Port 1 und der Port 2 nicht in macOS. Das sind beide an der Front des

Gehäuses beide 2.0. Auch Geräte die ich an Port 2 anschließe werden am Port 1 angezeigt.

Beitrag von „LetsGo“ vom 14. Dezember 2021, 21:13

[plutect](#)

Versuchs mal mit einer Maus oder Keyboard

- For old computers with OHCI/UHCI and EHCI controllers, you will need to plug in a mouse/keyboard to map the USB 1.1 personalities, as most USB 2 devices will end on the USB 2 personality.

Falls das Mapping funktioniert solltest du definitiv die Kombi aus UTBMap.kext+USBToolBox.kext verwenden, da du dir das Renaming des EHCI Controller sparst. Zumindest verstehe ich das so.

Features

- Attach to the controller instance or parent device, allowing for more ways to match
- Ignore port definitions from ACPI to force macOS to enumerate all ports manually
 - Bypasses borked ACPI as seen on some Ryzen motherboards and 400 series Intel motherboards
 - Replaces SSDT-RHUB
- Override any built-in Apple USB maps attaching based on SMBIOS and controller name
 - **Removes the need for controller renames in ACPI patches**
- Does not require model identifier specified in USB map (if attaching to PCI device)
- Very compatible with existing USB maps (port format is the same)
- Does not hardcode any port maps, unlike USBInjectAll

This does **not** patch the port limit.

EDIT: Nochmals durchgesehen:

Falls die beiden physischen Frontports vom EHCI Controller an einem internen USB 2 Port hängen, ist das wie ein interner Hub zu handhaben. Demnach wäre der auch als Type A: 0 zu deklarieren und nicht als Internal:255.

EDIT 2:

Da du eigentlich bei keinem deiner Controller das Portlimit von 15 überschreitest, könntest du ja gleich im "Select Ports and Build Kext" Menü mit "Select All (A)" alle Ports aktivieren und dann den UTBMap.kext erstellen.

Nur eben den Connector-Typ würde ich von 255 auf 0 setzen, da es sich bei den Frontports nicht um interne Geräte handelt.

Beitrag von „user232“ vom 15. Dezember 2021, 08:23

Man könnte sich mittels Windows, wie im Eingangspost beschrieben, einen **UTBMap.kext** erstellen und dann in das System einpflegen. Neu booten, die [Feinkonfiguration](#) mit Hackintool absolvieren, da übersichtlicher, dann den altbewährten USBPorts.kext generieren?

Edit:

Hab ich mal gemacht, nutze jetzt den mit Hackintool neu generierten USBPorts.kext, mal sehn wie es sich so verhält.

Beitrag von „LetsGo“ vom 15. Dezember 2021, 17:49

[user232](#)

Eigentlich gibt es kein "Feintuning" mit Hackintool. Entweder die Ports wurden richtig gemappt

oder nicht. Egal, ob man USBToolBox, Hackintool oder das USBMap Tool vom Dortania Guide nimmt. Der Vorteil von USBToolBox ist, dass wenn man Windows installiert hat und zusätzlich BigSur 11.3+ benutzt (wo der XhciPortLimit Quirk nicht mehr greift) so das Mapping am einfachsten ist.

Außerdem nutzt der USBPorts.kext, den man mit Hackintool erstellt als IOClass AppleUSBMergeNub, im Gegensatz zum USBMap.kext bzw. UTBMap.kext, die als IOClass AppleUSBHostMergeProperties nutzen. Und im Dortania Guide ist beschrieben, dass AppleUSBMergeHub zu Sleep Problemen bei Catalina 10.15 und höher führen kann. Kann und nicht muss!

<https://dortania.github.io/Openelec/sal/sleep.html#fixing-usb>

Beitrag von „kaneske“ vom 15. Dezember 2021, 19:00

Ich empfehle ja ungern Computerblöd aber ich mache das nicht mehr mit nativen Installationen.

Windows PE Stick fertig machen, USBToolbox Tool drauf kopieren und davon booten: aus dem BIOS den Stick als Bootvolume verwendet.

Startet mit schnellem Stick sehr fix, danach Ports mappen, kext liegt dann auf dem Stick. In die EFI einbinden und fertig.

Schont die Nerven und das Zeitmanagement.

<https://www.computerbild.de/artikel/k-erstellen-16583331.html>

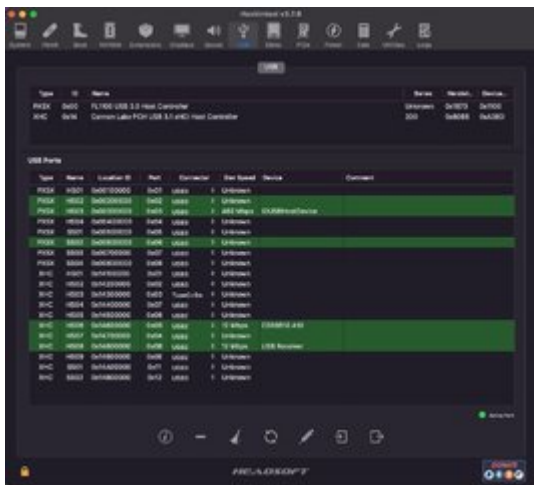
Beitrag von „user232“ vom 15. Dezember 2021, 19:20

Hat ja bei meinen zwei Hacks mittels Windows und dem **UTBMap.kext** problemlos

<https://www.hackintosh-forum.de/forum/thread/55483-usb-ports-mappen-unter-windows/>

funktioniert, 15 Portlimit gibts bei meinen Systemen nicht. Hab dann einfach mal mit den Kexten etwas rumgespielt und da kam mir der Gedanke, das gute Hackintool mal wieder einzusetzen und neuen USBPorts.kext generiert und ola alles funktioniert wie mit dem UTBMap.kext auch sleep, ich lass den mal ne Weile im System.

Was mit aufgefallen ist USBToolBox (Windows) hat meinen zweiten Controller (USB-3.0 PCIe-Steckkarte) erkannt, das hatte Hackintool nie fertig gebracht, also auch vor 11.3. und die Ports sind bei XHC nicht aufsteigend, d.h. nach 0x03 kommt 0x07, nicht 0x04. Beim PXSX-Controller stimmt die Reihenfolge.



Beitrag von „LetsGo“ vom 15. Dezember 2021, 19:21

[kaneske](#)

Wahrscheinlich die beste Lösung für jene, die kein Dual Boot System nutzen um die Ports schnell und sauber zu mappen.

[user232](#)

Das 15 Port Limit/Controller gibt dir aber MacOS vor.

Das mit der Nummerierung der Ports ist nur Kosmetik. Hackintool beläßt es bei der Originalnummerierung, wie sie in der USB-SSDT festgelegt sind. USBToolbox nummeriert sie von 0x01-0x0f der Reihenfolge nach nochmals neu.

Oder vielleicht verständlicher, wenn man sich ein Mapping von USBToolbox im Hackintool anschaut: Die Location ID ist schön der Reihe nach sortiert, in der Port Spalte sieht man die Originalnummerierung, wie sie in der USB-SSDT festgelegt sind. Bei dir ist z.B der HS06 der Port 9 in der SSDT oder halt in USBToolbox.

Beitrag von „user232“ vom 15. Dezember 2021, 19:50

Ich wollte damit sagen, dass ich gar nicht mehr als 15 Ports physikalisch zur Verfügung habe. Welche USB-SSDT meinst du, ich finde die nirgends 😊

Hab auch mal in der Info.plist von USBToolbox nachgeschaut, nada.

Beitrag von „LetsGo“ vom 15. Dezember 2021, 20:08

Dein PC hat die Informationen über deine USB Ports in einer SSDT oder auch DSDT gespeichert. Daher beziehen die Tools wie z.B: Hackintool, USBToolBox die Informationen. Der RHUB des XHCI Controllers wird mit 0x1400000 adressiert somit erhält der erste Port die Adresse 0x14100000, der zweite 0x14200000 bis eben zum Maximum eines XHCI Controllers von 26 Ports 0x141A0000. MacOS hat ein Portlimit von 15/Controller.

Du musst beim Portlimit aber bedenken, dass es sich nicht um physikalische Ports handelt sondern auch die Abwärtskompatibilität berücksichtigt. D.h: Jeder USB 3 Port (Connector Typ 3) beansprucht 2 Ports.

So würden 7 physikalische USB 3 Ports bereits 14 Ports beim Mapping beinhalten.

Der Begriff USB-SSDT war jetzt eher allgemein gehalten und wird auch anders heißen. Bei mir z.B: einfach SSDT14.dsl (aml). Siehe Anhang.

Lies dir [das](#) mal durch. Ist hervorragend beschrieben.

Beitrag von „user232“ vom 15. Dezember 2021, 20:18

Das mit den zwei Ports und USB 3.0 etc ist mir schon bekannt.

Wo befindet sich bei dir diese ssdt14.dsl?

Beitrag von „LetsGo“ vom 15. Dezember 2021, 20:24

In Windows wird z.B. beim Ausführen des Programms [QtiASL](#) ein Ordner ACPI auf dem Desktop erstellt, die diese SSDT enthält. In MacOS funktioniert mit MaciASL unter dem Punkt "New from ACPI". Dort sollte diese SSDT auch zu finden sein.

Beitrag von „bluebyte“ vom 15. Dezember 2021, 20:29

[kaneske](#) Macrium Reflect läuft auch mit Windows PE. Das Tool lässt sich von dort auch gut nutzen.

Beitrag von „user232“ vom 15. Dezember 2021, 20:59

Ich hab jetzt mal alle 13 SSDTs unter "New from ACPI" durch geklickt, von USB hab ich jetzt

nichts gefunden, vielleicht gehn mir auch schon die Augen über, aber lass mas gut sein, war nur so ein Gedanke, wo die Benennung der Ports herkommt, deine Informationen waren für mich hilfreich, Danke dir. 😊

Beitrag von „LetsGo“ vom 15. Dezember 2021, 21:22

Manchmal sind die Ports direkt in der DSTD beschrieben.

Beitrag von „kaneske“ vom 15. Dezember 2021, 21:23

[Zitat von bluebyte](#)

[kaneske](#) Macrium Reflect läuft auch mit Windows PE. Das Tool lässt sich von dort auch gut nutzen.

kann ich damit auch die USB ports mappen?

Beitrag von „EdD1024“ vom 9. April 2022, 09:17

[Zitat von kaneske](#)

Ich empfehle ja ungern Computerblöd aber ich mache das nicht mehr mit nativen Installationen.

<https://www.computerbild.de/ar...k-erstellen-16583331.html>

Fängt schon mal mit Zählen an:

Zitat

Es sind zwei Zutaten nötig, um ein Notfall-Windows für den USB-Stick zu erstellen: [Win10PE SE](#), eine [Windows-10-Installationsdatei](#) und [Rufus](#).

Lange nicht mehr so gelacht...

Beitrag von „kaneske“ vom 9. April 2022, 09:43

Und was soll das nun aussagen?

[EdD1024](#)

Beitrag von „EdD1024“ vom 9. April 2022, 11:03

Bestätigt deine Einstellung zu CB...

Beitrag von „user232“ vom 9. April 2022, 22:37

[EdD1024](#) 🙌, und sonst alles klar?

Beitrag von „KingBaggy1997“ vom 2. Februar 2023, 16:00

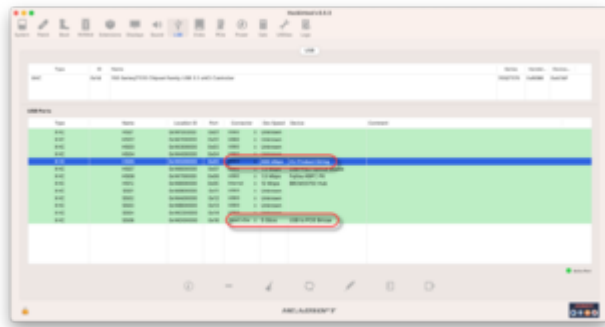
Wie bindet man denn unter Windows die Kext. in der Config List ein ?

Beitrag von „Max“ vom 2. Februar 2023, 16:19

Kannst du mit beispielsweise Propertree machen

Beitrag von „Arkturus“ vom 16. Februar 2023, 18:14

ich krieg das Portmapping an meinem KBL-Desktop nicht korrekt hin. Es bleibt der USB3.0 - Port übrig, der nur mit 480 Mbps angezeigt wird. Möglicherweise ist HS05 der zugehörige USB2 Port, aber egal was ich mache, es gibt an dem rückseitigen Port keine Ausgabe mit 5Gbps.



Der Kext nach Finetuning mit Hackintool

[USBMapPorts-KBLD-HT.kext.zip](#)

der Kext nach Portmapping mit Windows.exe vom 20.11.v.J.

[UTBMap-KBL-DESKTOP.kext.zip](#)

Kann man hier anhand der info.plist das Problem lösen?

EDIT: Nunmehr der 3. Versuch heute gescheitert, einen neuen USBMap.kext zu generieren.

Ergebnis sieht so aus. Nicht zu gebrauchen, fast alles auf intern und die meisten Ports funktionieren überhaupt nicht.



Am T460 klappt das hingegen tadellos.



Beitrag von „KungfuMarek“ vom 16. Februar 2023, 18:40

Ich hatte auch mal das Problem mit einem Port, der wollte einfach keine 5Gbps bereitstellen.

Habe dann mal aus Spass auf Typ C gemappt und dann ging der Port auch mit 5Gbps.

Macht absolut keinen Sinn, aber läuft.

Versuch es mal! [Arkturus](#)

Edit:

Sehe gerade, dass es der HS05 ist.

HSxx Ports sind immer nur USB2.

Hast du das Mapping unter Windows neuerstellt oder bist du da vorgegangen?

Beitrag von „Arkturus“ vom 16. Februar 2023, 18:45

Ja, das hatte ich aber bereits. Ich hatte erst in Erinnerung, dass der Port HS05 neben dem SS06 USB-C wäre, aber lt. Handbuch ist es USB3.0, deshalb hatte ich den schon auf Typ-C eingestellt. Trotzdem nur 48Mbps. [KungfuMarek](#)

Ich machs nochmals auf TypC+SW.

Habe TypC und TypC+SW probiert. Klappt nicht.

Alle anderen USB3.0 und der TypC sind SS0x, der eine ist HS05 und soll USB3.0 sein, ist blau und lt. Manual auch USB3.0

EDIT:

Ja, unter Windows. Das Mapping ist schon von November, im Moment kriege ich das nicht wiederholt.

Ich muss mal in den alten Kext vom Catalina-Mapping schauen.

EDIT: Catalina hatte überhaupt nur zwei USB3.0 gemappt. Hier mal die Info.plist

Dann ist das eben so. Auch kein Drama, wenn man es weis. Da stecke ich den Drucker dran und gut.

Beitrag von „N0b0dy“ vom 16. Februar 2023, 19:09

es fehlt SS05

da geht um Ausgang neben USB-C mit HS05/SS05, auf deinem Screenshot HS05 da aber SS05 fehlt.

Außerdem alle Ausgänge, die HSXX/SSXX gleichzeitig sind, müssen als USB3.0 deklariert.

Beitrag von „KungfuMarek“ vom 16. Februar 2023, 19:31

Bei meinem z690 Mainboard passen die HSxx/SSxx nicht von der Nummerierung zusammen. Als Beispiel HS04/SS06.

Aber N0b0dy könnte da recht haben, das ist ja von Board zu Board unterschiedlich, bei meinem X299 passten die Nummerierungen jedoch schon.

Denke auch, dass dir ein SSxx Port fehlt. Deswegen würde ich das Mapping nochmal neu machen unter Windows. Falls du das wiederholen kannst.

Wenn ich den Kext im Editor bearbeite, dann ist mein Mapping komplett falsch und durcheinander, geht nur wenn ich über Windows mappe.

Beitrag von „Arkturus“ vom 16. Februar 2023, 19:35

Problem gelöst. SS05 ist wahrscheinlich im Finetuning mit HackinTool verloren gegangen. Im Masterkext noch vorhanden gewesen. Hätte ich sehen müssen, aus der Erinnerung war SS05 weiß, aber er ist belegt.

Danke für die zielführenden Hinweise. [KungfuMarek](#) N0b0dy

Alles funktioniert:

