

Einleitung

Dieser Beitrag ist eine Übersicht der unterschiedlichen Methoden, die mir zum USB-Mapping bekannt sind.

Welche Methode des USB-Mappings ist vorzuziehen?

Es gibt verschiedene Methoden die USB Ports zu mappen, um diese im MacOS verwenden zu können. Mir sind bisher drei bekannt. Prinzipiell geht es bei allen Methoden so, dass die App bzw. das Skript zuerst alle hardwaremäßig vorhandenen USB-Ports erkennen (können) muß, bevor man dies einschränkt, um im 15-Port Limit pro Controller zu bleiben. Während man mit einer Mac-Methode (Python-Skript oder Hackintool) zuerst den Mac aufwendiger darauf vorbereiten muß (incl. einiger Reboots und Anpassungen an der OC Config), hat man es da unter Windows OS wesentlich einfacher, da da alle Controller / USB-Ports gleich erkannt werden. Zudem kann die Windows App auch den USB-2/3 Zusammenhang der Ports aufzeigen, was einem den Abgleich mit dem Mainboard-Manual und letztlich die Entscheidung leicht macht, auf welchen Port man verzichten kann. Also, wann immer möglich, **verwende die WIN-Methode**.

Warum das Ganze?

Das MacOS hat seit "El Capitan" (v10.11) ein USB-Port Limit von 15 pro Controller eingeführt, was aus Apple Sicht durchaus Gründe hat (mehr dazu im Hintergrundwissen). Alles was diese Port-Anzahl übersteigt *wird nicht erkannt*. Zudem kann dies mit jedem System-Upgrade (möglicherweise) anders von Apple gehandhabt werden. Also bei Problemen dies checken bzw. vor einem MacOS Upgrade sich zuerst informieren. :)

Zum besseren Verständnis

Die ganze Problematik entsteht, weil technisch ein **USB-3 Port** aus insgesamt **zwei** USB Ports besteht - mit einem USB-2 Anteil und dem USB-3 Anteil - und damit auf modernen PC das Limit schnell überschritten ist. Sollte z.B. die vorhandene Hardware 2 x USB 2.0 Anschlüsse und 7 x USB 3 Ports haben, wurde der Port-Limit bereits überschritten, da die Port-Anzahl: $7 \times 2 = 14 + 2 = 16$ beträgt und damit um ein Port über dem Limit ist.

Der eigene **USB Kext Injektor** sorgt dafür, dass nur die Ports im MacOS auftauchen, die erstens tatsächlich genutzt werden, zweitens deren Typen richtig beschrieben sind und drittens im Limit von **15 Ports pro Controller** bleiben. Sollten die Anschlüsse das **Port Limit von 15 Ports pro Controller** sprengen (was sehr wahrscheinlich ist), müssen Ports getrennt werden. Entweder dabei mehrere USB Anschlüsse **komplett deaktivieren** oder von einem USB 3.0 Port **den USB 2.0 oder den 3.0 Anteil wegnehmen**.

Methode 1: USB- Mapping im Windows OS

A. Die Schritte zum USB-Mapping

Die Infos habe ich aus dem Hackintosh-Forum zusammen getragen.

Das USB-Ports Mapping lässt sich unter Windows OS mit einem Open Source Programm namens: [Windows.exe](#) (es gibt zwar auch eine MacOS Version, doch von der raten selbst die Entwickler ab) leichter erledigen, als mit der Hackintool App (bzw. Python Skript).

Bei vorhandener WIN-Partition ist diese Methode sehr hilfreich, da man normalerweise schon alle USB-Ports verfügbar sind. Die Injektor-Erstellung bietet zwei Varianten an. Eine die auf die nativen Apple KEXTe setzt, die andere ersetzt diese. Habe beides getestet, wobei ich letztlich beim einzelnen Injektor-KEXT: **USBMap.kext** blieb, obwohl die Kombination von *UTBMap.kext* & *USBToolBox.kext* es natürlich auch macht. Da ich keinen Unterschied feststellen konnte, blieb es bei der "1-KEXT-Variante".

Ab **Big Sur 11.3** müssen die USB-Ports gemappt sein. Das Programm **Hackintool** funktioniert ab da **nicht** mehr, eine Methode unter Windows wird hier erläutert. (Angelehnt an den [Hackintosh-Beitrag](#) von [LetsGo](#)).

Mit dieser Methode ist man auch von **SMBIOS** unabhängig.

USB-Ports mappen unter Windows - Beispiel mit dem Injektor(en): **UTBMap.kext** (erfordert zusätzlich den **USBToolBox.kext**) - Standard-Einstellung des Programms. Mit **Option N** (= *Enable*) kann man auf die "1-KEXT-Variante": *USBMap.kext* umstellen.

Nun zu den Schritten des USB-Mappings (Anm.: *Drücken* meint Option eingeben und mit *Enter* bestätigen):

1. **Download** von Github: [Windows.exe](#)
2. Entzippe in ein neues Verzeichnis und **starte** darin die App: *Windows.exe*
3. *Optional* - Drücke **Taste N** (= *Use Native Classes*) um auf USBMap statt UTBMap umzustellen.
4. Drücke **Taste D** (= *Discover Ports*)
5. **Stecke nacheinander in jeden USB-Port einen USB3-Stick - warten bis dieser vom Programm erkannt wird!**
6. Wenn alle USB-Ports bearbeitet sind, drücke die **Taste B** (= *Back*)
7. Drücke die **Taste S** (= *Select Ports and Build Kext*)
8. Drücke die **Taste P** (= *Enable All Populated Ports*)
9. Drücke die **Taste K** (= *Build UTBMap.kext*) [bzw. *USBMap.kext*, falls darauf eingestellt. Nachfolgende Punkte sind dann natürlich verändert. Punkt 11 entfällt und bei 12 die *config.plist* daran anpassen!]
10. **UTBMap.kext** wird erstellt
11. Der *UTBMap.kext* muss in Verbindung mit dem [USBToolBox.kext](#) (hat einen *executable path* - in der *config.plist* beachten!) verwendet werden. Bei Verwendung dieser Kombi sollte SSDT-RHUB ebenfalls überflüssig sein.
12. Das erstellte *UTBMap.kext* und das *USBToolBox.kext* in */EFI/OC/Kexts* kopieren und die **config.plist** wie folgt bearbeiten und danach einen Reboot durchführen:

> Item 6	Dictionary	◇ 8 items
▼ Item 7	Dictionary	◇ 8 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	◇ 8 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	◇ 8 items
> Item 10	Dictionary	◇ 8 items
> Item 11	Dictionary	◇ 8 items

B. Die Mapping-App in der Verwendung

Zum Anfang sei angemerkt: Mach dich mit dem Programm vertraut. Teste die Optionen aus, sehe die Veränderungen, dann wird vieles in dieser Erklärung klarer. Keine Angst, man kann nichts kaputt machen.

```

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:

```

Die Bedienung funktioniert mit dem **jeweiligen Buchstaben** gefolgt von **Enter**.

z.B: **C** und **Enter** für das -> **Change Settings** Menü

```
#####
#                               #
#                               #
#####

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:
```

Das Settings Menü (C):

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

- **T. Show Friendly Types** bedeutet mit:
Disabled = die Port-Typen werden nur mit **Zahlen** gekennzeichnet,
Enabled = die Port-Typen werden ausgeschrieben.
 Beispiel: aus **3** wird der **USB-3 Type A**, aus **9** wird **USB-3 Type C**, usw.
- **N. Use Native Classes** bedeutet mit:
Disabled = der **UTBMap.kext** wird erstellt (benötigt zusätzlich den **USBToolBox.kext**),
Enabled = der **USBMap.kext** wird erstellt (d.h. die nativen Apple KEXTe werden genutzt)
- **A. Add Comments to Map** bedeutet mit:
Disabled = keine Beschreibung des USB Ports. Lässt sich später noch hinzufügen.
Enabled = die Beschreibung (siehe *Set Custom Names*) wird als *comment* in die *info.plist* des erstellten Kextes übernommen.
- **C. Bind Companions** bedeutet mit:
Disabled = keine Erkennung des USB-2/3 Zusammenhangs eines Ports.
Enabled = automatische Erkennung des USB-2/3 Zusammenhangs eines Ports. Da ein physischer USB-3 Port wegen der Abwärtskompatibilität einen USB-3 und USB-2 Teil besitzt, nützt diese Funktion beim Mappen. Einfach USB-3 oder 2 Stick in den Port stecken und der Gegenpart wird automatisch erkannt.

```
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
```

Im Gegensatz zu Port-Types als Zahl:

```
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
```

Beispiel: Mit **Show Friendly Types = Enable**
 - Port 1 mit **Type 3** wird zu **USB-3 Type A**,
 - Port 2 mit **Type 9** wird zu **USB-3 Type C**

```

C:\Programme (portable)\USB > +
#####
#                               #
#####
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:

```

Discover Ports (D):

Als Beispiel wurde ein USB Stick in den Front **USB-3 Port (Port 17)** gesteckt, um den weiteren Verlauf und die Funktionen des Programms zu erklären.

```

C:\Programme (portable)\USB > +
#####
#                               #
#####
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 USBMap.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:

```

Dann mit **(B)**ack und in das "*Select Ports and Build Kext*" Menü wechseln:

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.

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!](#)


```

#####
# 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 entry: 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 parts, they do not have a switch (type 10).

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

B. Back
Select an option:

```

- 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

Nun zu diesen Punkten:

-**Select Ports ...**: dient zum an-/abwählen der Ports.

-**Change Types**: Würde das Programm beispielsweise einen Port falsch deklarieren, ließe sich hiermit den Typ ändern.

In diesem Beispiel würde die Eingabe: **t:17:0** den **Port 17 und 6** ändern von:
Typ 3 (USB-3 Typ A) -> Typ 0 (USB-2 Typ A)

- 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

```

- 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

-**Set custom names ...**:

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

```

#####
# 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: |

```

Wie es jetzt anhand *dieses Beispiels* das System aussehen sollte:

1) Discover Ports: Stick in alle benötigten Ports stecken!

```

#####
#          Select Ports and Build Kext          #
#####

Intel(R) USB 3.1 eXtensible-Hostcontroller - 1.10 (Microsoft) | USB 3.0 (XHCI) | 16/26 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)
    - DataTraveler 3.0 - operating at USB 2.0
[#] 10. Port 10 | USB 2.0 | Type 3 (guessed) | Companion to 21
[#] 11. Port 11 | USB 2.0 | Type 0 (guessed)
    - DataTraveler 3.0 - operating at USB 2.0
[#] 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
    - DataTraveler 3.0 - operating at USB 3.0
[#] 18. Port 18 | USB 3.0 | Type 9 (guessed) | Companion to 2
    - Ultra USB 3.0 - operating at USB 3.0
[ ] 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
    - DataTraveler 3.0 - operating at USB 3.0
[#] 22. Port 22 | USB 3.0 | Type 3 (guessed) | Companion to 1
    - DataTraveler 3.0 - operating at USB 3.0
[#] 23. Port 23 | USB 3.0 | Type 3 (guessed) | Companion to 12
    - DataTraveler 3.0 - operating at USB 3.0
[#] 24. Port 24 | USB 3.0 | Type 3 (guessed) | Companion to 8
    - DataTraveler 3.0 - operating at USB 3.0
[ ] 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:

```

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

```

- 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]

```

3) Um das **Portlimit von 15 einzuhalten deaktiviere** die **Companion Funktion** im *Settings Menü* und wähle den **USB-2 Teil** (Port 2) vom **Front USB-C Port** ab.

```

#####
#          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 | 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)
    - DataTraveler 3.0 - operating at USB 2.0
[#] 10. Port 10 | USB 2.0 | Type 3 (guessed) | Companion to 21
[#] 11. Port 11 | USB 2.0 | Type 0 (guessed)
    - DataTraveler 3.0 - operating at USB 2.0
[#] 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
    - DataTraveler 3.0 - operating at USB 3.0
[#] 18. Port 18 | USB 3.0 | Type 9 (guessed) | Companion to 2
    - Ultra USB 3.0 - operating at USB 3.0
[ ] 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
    - DataTraveler 3.0 - operating at USB 3.0
[#] 22. Port 22 | USB 3.0 | Type 3 (guessed) | Companion to 1
    - DataTraveler 3.0 - operating at USB 3.0
[#] 23. Port 23 | USB 3.0 | Type 3 (guessed) | Companion to 12
    - DataTraveler 3.0 - operating at USB 3.0
[#] 24. Port 24 | USB 3.0 | Type 3 (guessed) | Companion to 8
    - DataTraveler 3.0 - operating at USB 3.0
[ ] 25. Port 25 | USB 3.0 | Type 255 (guessed)
[ ] 26. Port 26 | USB 3.0 | Type 255 (guessed)

Binding companions is currently off.

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

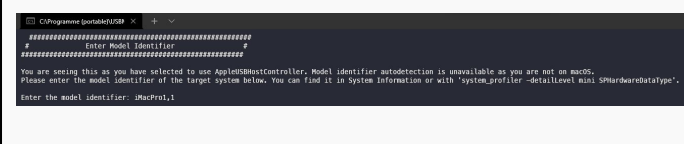
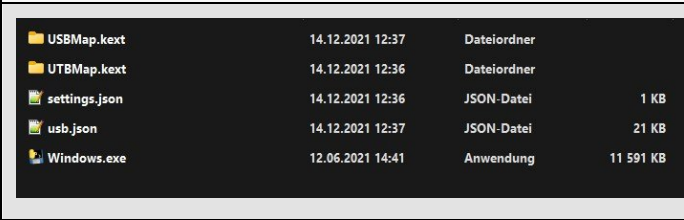
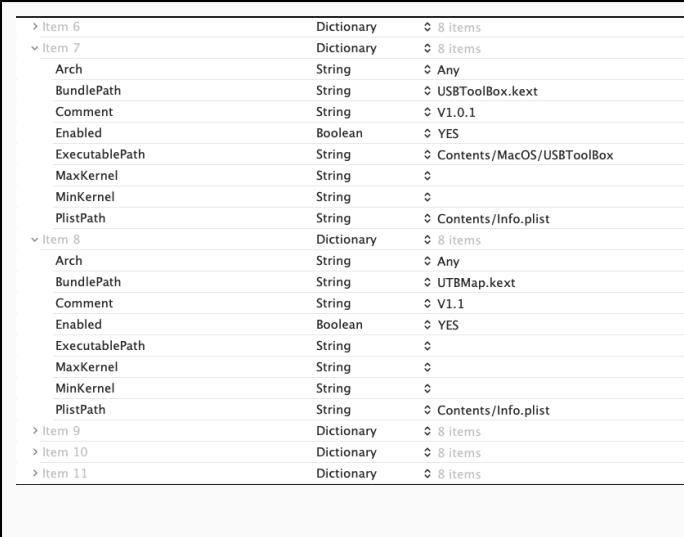
- Select ports to toggle with comma-delimited lists (eg. 1,2,3,4,5)
- Change types using this formula T:1,2,3,4,5:t where t is the type
- Set custom names using this formula C:1:Name - Name = None to clear
Select an option:

```

Dazu einfach **2** eingeben und *Enter* drücken (wäre *Bind Companions* aktiviert, würde damit Port 2 und 18 deaktiviert werden)

4) Nun erstelle den **USBMap.kext** (bzw. die zwei Kexte: *UTBMap.kext* und *USBMap.kext*) mit **(K) Enter**.

Einfach im Settings Menü (*Use Native Classes*) umstellen.

	<p>Bei Erstellung des <i>USBMap.kext</i> wird noch nach dem SMBIOS gefragt: ich nutze z.B: <i>iMacPro1,1</i></p>
	<p>5) Die jeweiligen generierten Kexte befinden sich im Verzeichnis: <i>USBToolBox</i></p>
	<p>6) Benutzt man <i>UTBMap.kext</i>, muss noch zusätzlich der <i>USBToolBox.kext</i> von Github runtergeladen werden. Dann Beide in den Kext-Ordner der EFI kopieren und in die <i>config.plist</i> einbinden.</p> <p>Beachten, dass der <i>USBToolBox.kext</i> einen ausführbaren Pfad (<i>ExecutablePath</i>) hat.</p> <p>Für den <i>USBMap.kext</i> wird <i>USBToolbox.kext</i> <u>nicht</u> benötigt, da dieser die nativen Apple Kexte benutzt.</p>

Methode 2: USB Port Mapping mit Python Skript

Auf GitHub wird ein **Python-Skript** angeboten, mit dem sich im MacOS die **Zuordnung der USB-Ports** und die Erstellung des **benutzerdefinierten Injektor-Kext** durchführen lässt. (-> [Original-Artikel auf GitHub](#)).

Das **Python-Skript** ermöglicht die Zuordnung der USB-Ports und erstellt daraus einen eigenen Injektor-Kext namens -> ***USBMap.kext***.

Die Eigenschaften des Skripts

- Keine Abhängigkeit von USBInjectAll
- Kann XHCI- (Chipsatz, Drittanbieter und AMD), EHCI-, OHCI- und UHCI-Anschlüsse zuordnen
- Kann USB-2-HUBs zuordnen (erfordert, dass der übergeordnete Anschluss des HUBs Typ 255 verwendet)
- Entspricht dem Klassennamen, nicht dem Port- oder Controller-Namen
- Ermöglicht das Festlegen von *Spitznamen* für die zuletzt gesehenen (belegten) Ports bei der Erkennung
- Aggregiert angeschlossene Geräte über die Sitzungs-ID anstelle der fehlerhaften Port-Adressierung
- Kann anhand von Best-Guess-Ansätzen ACPI generieren, um Controller umzubenennen oder RHUB-Geräte bei Bedarf zurückzusetzen

A. Installation des Skripts

Mit Git:

Führe folgende Befehle im Terminal aus:

```
git clone https://github.com/corpnewt/USBMap
cd USBMap
chmod +x USBMap.command
```

Führe dann entweder mit `./USBMap.command` oder durch Doppelklick auf `USBMap.command` das USB Mapping durch. Eine funktionierende Python-Installation sollte natürlich vorhanden sein.

Ohne Git:

Die neueste Zip-Version des Repos [von GitHub](#) herunterladen. Entzippen und dann **USBMap.command** mit einem Doppelklick ausführen.

B. Anwendung des Skripts - den USBMap Injektor erstellen

Die einzelnen Schritte um den USBMap Injektor zu erstellen:

1. Vergewissere dich mindestens einmal das Menü: **D = Discover Ports** aus dem Hauptmenü von **USBMap.command** ausgeführt zu haben, damit das Skript weiß, welche USB-Controller vorhanden sind.
2. Wähle: **K = Create Dummy USBMap.kext** über das Hauptmenü von USBMap.command aus.
3. Fügen diesen **USBMap.kext** Dummy-Injektor dem **OC -> Kexts** Ordner zu und trage ihn ein in:
config.plist -> Kernel -> Add
4. **Rebooten**
5. Gehe in USBMap's **D = Discover Ports** und schließe ein **USB 2** und ein **USB 3** Gerät an jeden Port an - lasse das Skript sich **zwischen jedem Anschluss aktualisieren**.
6. Verwende **USBMapInjectorEdit.command**, um alle nicht unbedingt erforderlichen Ports zu deaktivieren (alle 15 Ports, die keine Tastatur/Maus/etc. sind und für die Grundfunktionen benötigt werden)
7. **Rebooten**
8. Gehe in USBMap's **D = Discover Ports** und schließe ein USB 2 und ein USB 3 Gerät an jeden Port an - lasse das Skript sich zwischen jedem Anschluss aktualisieren.
9. Gehe dann in das Menü: **P = Edit & Create USBMap.kext** und ändere die Typen so, dass sie mit den physischen Porttypen übereinstimmen, und schalte die gewünschten Ports (bis zu 15) um.
10. Erstelle die endgültige **USBMap.kext** und ersetze damit den Dummy-Injektor im **OC -> Kexts** Ordner

Methode 3: USB Port Mapping mit dem Hackintool

Soll ab Big Sur 11.3 nicht mehr richtig funktionieren! Wieso? Keine Ahnung - habe es noch nicht eruiert.

Zusatz für OpenCore: *USBInjectAll.kext* muss nicht verwendet werden. Es reicht aus, wenn **Kernel > Quirks: XhciPortLimit** auf **Yes/True** gestellt wird (gilt nur für neuere Boards).

Wenn dann die erstellte *USBPort.kext* verwendet wird, muss dieser **Quirk wieder auf NO/False** gestellt werden.

Methoden Hackintool:

Das Hackintool starten und den *USB-Tab* anklicken. Alle vorhandenen USB Anschlüsse werden darin angezeigt.

- Diese der Reihe nach testen - erst mit einem reinen(!) USB 2.0 Device, danach nochmals mit einem USB 3.0 Device, damit man sieht, welche in Benutzung sind.
- Die Ports aus der Liste löschen, welche nicht genutzt werden (= alle die nicht grün sind).
- Dann die Art des Anschlusses bestimmen:
 - USB 2.0 Anteil eines USB 3 Ports wird auf **USB3** gesetzt
 - USB 3.0 Anteil eines USB 3 Ports wird auf **USB3** gesetzt
 - Reine USB 2.0 Anschlüsse auf **USB2**
 - Besonderheit bei Typ-C: Gleicher Port in beide Richtungen = TypeC + SW; unterschiedlicher Port je nach Richtung = TypeC
 - Interne USB Ports (z.B. internes Bluetooth, Lüftersteuerung etc.) wird auf Internal gesetzt

Sollte die Anzahl der Anschlüsse das *Port Limit von 15 pro Controller* überschreiten, muss man sich von Ports trennen!

Dazu selbst entscheiden, ob man einen oder mehrere USB Anschlüsse komplett deaktiviert oder von einem USB 3.0 Port den USB 2.0 oder den 3.0 Anteil wegnimmt.

Sobald alles fertig konfiguriert ist, die Daten exportieren. Hackintool generiert meistens 3 Dateien: **SSDT-EC**, **SSDT-UIAC** und **USBPorts.kext**.

Die Dateien werden in den EFI/OC Ordner verschoben:

1. SSDT-EC kommt nach ACPI/patched
2. SSDT-UIAC ist für die Verwendung mit USBInjectAll gedacht
3. Die **USBPorts.kext** ist eine **Standalone Lösung** und der optimale Weg (kommt nach EFI/OC/Kexts). Nutzt man diese, dann kann die **USBInjectAll.kext** (falls verwendet, siehe Anfangshinweis) gelöscht werden. Die **SSDT-UIAC** braucht man dann ebenfalls nicht.