

USB-Ports mappen unter Windows

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 durchgelesen:

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.