

Unter Verwendung von **Clover Configurator**, dem **Hackintool** und dem **USBInjectAll.kext...**

## ...USB Anschlüsse Patchen

**Vorweg:** Ab macOS 10.14.1 funktionieren die 'USB port limit patches' nicht mehr, somit werden nicht alle USB Anschlüsse mit ihren technischen Möglichkeiten bereitgestellt.

RehabMan hat seinen **USBInjectAll.kext** erweitert, um verschiedene 'Boot Arguments' (boot flags) für das Ausklammern diverser Anschlüsse zu ermöglichen, diese Funktion nutzen wir um mit der folgenden Methode einen eigenständigen Kext zu erstellen, der sowohl die USB-Anschlüsse eindeutig deklariert, als auch deren Eigenschaften festlegt.

**Anmerkung:** Ausser einem Keyboard und einer Maus sollten keine weiteren externen Geräte an vorhandenen USB-Ausgängen Angeschlossen sein. (Keyboard und Maus für gewöhnlich an USB2).

**1. Vorbereitung)** Monte Deine EFI mit dem **Clover Configurator**, öffne den Ordner CLOVER/kexts/Other. Kopiere den **USBInjectAll.kext** dort hinein.

**1b. DSDT Patches)** die **USBInjectAll.kext** benötigt im **Clover Configurator** folgende ACPI-Patches (Einträge): Menue '**ACPI**' -> unter '**DSDT**' -> '**List of Patches**'

'change XHC1 to XHC' (bei mir nicht vorhanden)  
'change EHC1 to EH01'  
'change EHC2 to EH02'

-> **sichern** -> **ummount EFI** -> **beenden** -> **Neustart**.

**2. Hackintool)** Öffne nach dem Neustart das **Hackintool** und schaue im Reiter '**Installed Kexte**', ob USBInjectAll erfolgreich installiert und geladen wurde Bsp. USBInjectAll: Yes (Release-0.7.1).

**2b. Optional weitere USB-Kext'e?)** Gehe zum Reiter '**USB**', um die vorhandenen '**USB Controller**' in der entsprechenden Liste zu prüfen. Je nach vorhandenem USB Controller muss/kann ggf. ein weiterer USB-Kext installiert werden:

- 8086:8CB1 und macOS < 10.11.1 = '**XHCI-9-series.kext**'
- 8086:8D31, 8086:A2AF, 8086:A36D, 8086:9DED = '**XHCIunsupported.kext**'
- 8086:1E31, 8086:8C31, 8086:8CB1, 8086:8D31, 8086:9C31, 8086:9CB1 = '**FakePCIID.kext**' + '**FakePCIID\_XHCIMux.kext**'

Installiere ggf. den/die entsprechenden Kext -> Neustart -> öffne **Hackintool** erneut.

**3. Hackintool)** Gehe zum Reiter -> '**USB**', markiere alle Elemente der USB Anschluss-Liste und drücke erst die '**entfernen**' Schaltfläche und danach die Schaltfläche '**Aktualisieren**'.

Zwei Ports, für Maus und Keyboard, müssten **grün** sein. Notiere dir die Namen (z.B. HS13, HS14).

**4. Boot Arguments USB2)** Um beim nächsten Start nur die Anschlüsse **USB2** zu laden, öffne Deine 'config.plist' mit dem **Clover Configurator**, gehe ins Menue '**Boot**' und trage unter '**Boot Arguments**' den Eintrag "-uia\_exclude\_ss uia\_include=HSxx,HSxx". **ACHTUNG:**

Ändere '**xx**' in die Anschluss-Namen für Maus und Tastatur (z.B. HS13,HS14), die Du im **Punkt 3.** im Hackintool als „Keyboard“ und „Mouse“ notiert hast.

-> **sichern** -> **ummount EFI** -> **beenden** -> **Neustart**.

**5. USB 2.0 Anschlüsse?)** Öffne das **Hackintool**, gehe zum Reiter -> '**USB**' • Schliesse ein USB 2.0 Gerät nacheinander an alle vorhandenen USB Anschlüsse deines Systems an (einstecken -> auswerfen), dadurch werden die gefundenen aktiven Anschlüsse **grün** hervorgehoben.

**5b. )** Markiere alle nicht grün hervorgehobenen Anschlüsse, drücke die '**entfernen**' Schaltfläche!  
Beende das **Hackintool**

**6. Boot Arguments USB3)** Um im nächsten Schritt nur die **USB3** Anschlüsse (mit Ausnahme **USB2** für Mouse und keyboard) zu laden Öffne erneut deine 'config.plist', gehe ins Menue '**Boot**' ändere bei '**Boot Arguments**' den Eintrag „-uia\_exclude\_ss“ in „-uia\_exclude\_hs“. Die „uia\_include=...“ für 'Mouse' und 'Keyboard' bleiben stehen!  
-> **sichern -> unmount EFI -> beenden -> Neustart.**

**7. USB 3 Anschlüsse?)** Starte das Hackintool, gehe zum Reiter -> '**USB**' • Schliesse ein USB 3.0 Gerät nacheinander an alle vorhandenen USB3 Anschlüsse deines Systems an • Schliesse ein USB TypeC Gerät an alle vorhandenen TypeC Anschlüsse an (in beiden Ausrichtungen), Dadurch werden die gefundenen aktiven Anschlüsse **grün** hervorgehoben.

**7b. )** Markiere alle nicht grün hervorgehobenen Anschlüsse, drücke die '**entfernen**' Schaltfläche!

**7c. Connector-Typ deklarieren)** Setze bei '**Connector**' über die Drop-Down-Liste für jeden Anschluss den passenden Connector-Typ. USB2 (HSxx) Anschlüsse die über eine USB3-Buchse verbunden/erkannt werden, sollten als '**USB3**' deklariert werden.

**TypeC:** Wenn bei verschiedener Ausrichtung der gleiche HSxx/SSxx Anschluss benutzt wird, enthält der TypeC Anschluss einen internen Umschalter und sollte als '**TypeC+Sw**' deklariert werden. Wenn bei verschiedener Ausrichtung unterschiedliche HSxx/SSxx Anschlüsse benutzt werden, enthält der TypeC Anschluss keinen internen Umschalter und sollte als '**TypeC**' deklariert werden.

USB-Anschlüsse mit dauerhaft verbundenen Geräten (z.B. Bluetooth aus WLAN/BT-Kombi-Karte) sollten als '**Internal**' deklariert werden. Interne USB-HUBs, welche standardmäßig über die ACPI-Geräte PR11 und PR21 erreichbar sind, sollten als '**Internal**' deklariert werden.

**7d. Kext & SSDT-EC.aml erstellen)** Benutze die '**Exportieren**' Schaltfläche um das Ergebnis als Datei auf deinem Schreibtisch zu speichern. Kopiere den **USBPorts.kext** nach /EFI/CLOVER/kexts/Other. Wurde eine **SSDT-EC.aml**-Datei erstellt, kopiere diese nach /EFI/ CLOVER/ACPI/patched.

**8. Nicht mehr benötigten Kext'e und Einstellungen bereinigt)** Entferne die zuvor gesetzten Boot-Flags (-uia\_exclude\_ss, -uia\_exclude\_hs, uia\_include=xx).

Entferne die Patches: Menue 'ACPI' -> 'DSDT' -> 'List of Patches' (XHC1, EHC1, EHC2)

Entferne den **USBInjectAll.kext** und ggf. in **2b)** weitere installierte USB-Kext'e.

-> **sichern -> unmount EFI -> beenden -> Neustart.**

**9. Ergebnis prüfen)** Öffne **Hackintool** und gehe zum Reiter -> '**USB**', markiere alle Elemente der USB Anschluss-Liste und wähle nacheinander die '**Entfernen**' und '**Aktualisieren**' Schaltfläche. Damit kannst du prüfen, ob alle vorher deklarierten Anschlüsse korrekt eingelesen wurden. Wenn du einen Connector-Typ mit dem **Hackingtool** ändern möchtest, musst du im Nachgang den **USBPorts.kext** wieder exportieren und den aktuell eingesetzten Kext ersetzen.

-> Sollte dir ein Fehler unterlaufen sein, lösche alle Ergebnisse und beginne von vorne!

Q. Was ist der USBPorts.kext?

A. Das ist eine Codeless Kernel Extension zur Deklaration der verschiedenen USB Anschlüsse

Q. Benötige ich dazu eine SSDT-UIAC.aml?

A. Nein, wird hier nicht benötigt, da die "codeless" Kext ausreicht

