

Erledigt

USB2 Devices -> EHCI Routing zu AppleUSBEHCI

Beitrag von „JoeHidden“ vom 22. Januar 2017, 20:39

Hallo in die Runde,

ich habe ein spezielles Problem.

Zunächst zum IST Stand:

Ich habe via DSDT die Anpassung von XHCI->XHC, EHCI->EH01 sowie ECH2-EH02 durchgeführt. Die neuste Version der USBInjectAll.kext kommt zum Einsatz und alle Devices werden von AppleUSBXHCIPCI bearbeitet. Die uia_exclude sind entsprechend gesetzt, es sind unter 15 Ports und es ist kein Patch der USB Treiber aktiv. Sowohl USB2 als auch USB3 Geräte werden korrekt erkannt und die Geschwindigkeit von USB3 ist korrekt. Im Prinzip entspricht alles der aktuellen [Anleitung](#) von [@kuckkuck](#).

Jetzt zum Problem:

Ich verwende ein externes USB Audio Device (ein UMC204HD von Behringer). Audio an sich ist ok, es kommt aber immer wieder zu Audioprobleme (Tonartefakte, Stottern, etc.). Das Problem ist dem Hersteller bekannt und liegt am Zusammenspiel mit dem XHCI Treibern bei Apple. Am MacBook via EHCI läuft das Interface ohne Probleme und die Gegenprobe - ein Focusrite Scarlett 2i2 von einem Freund zeigt das gleiche Verhalten am Hacki.

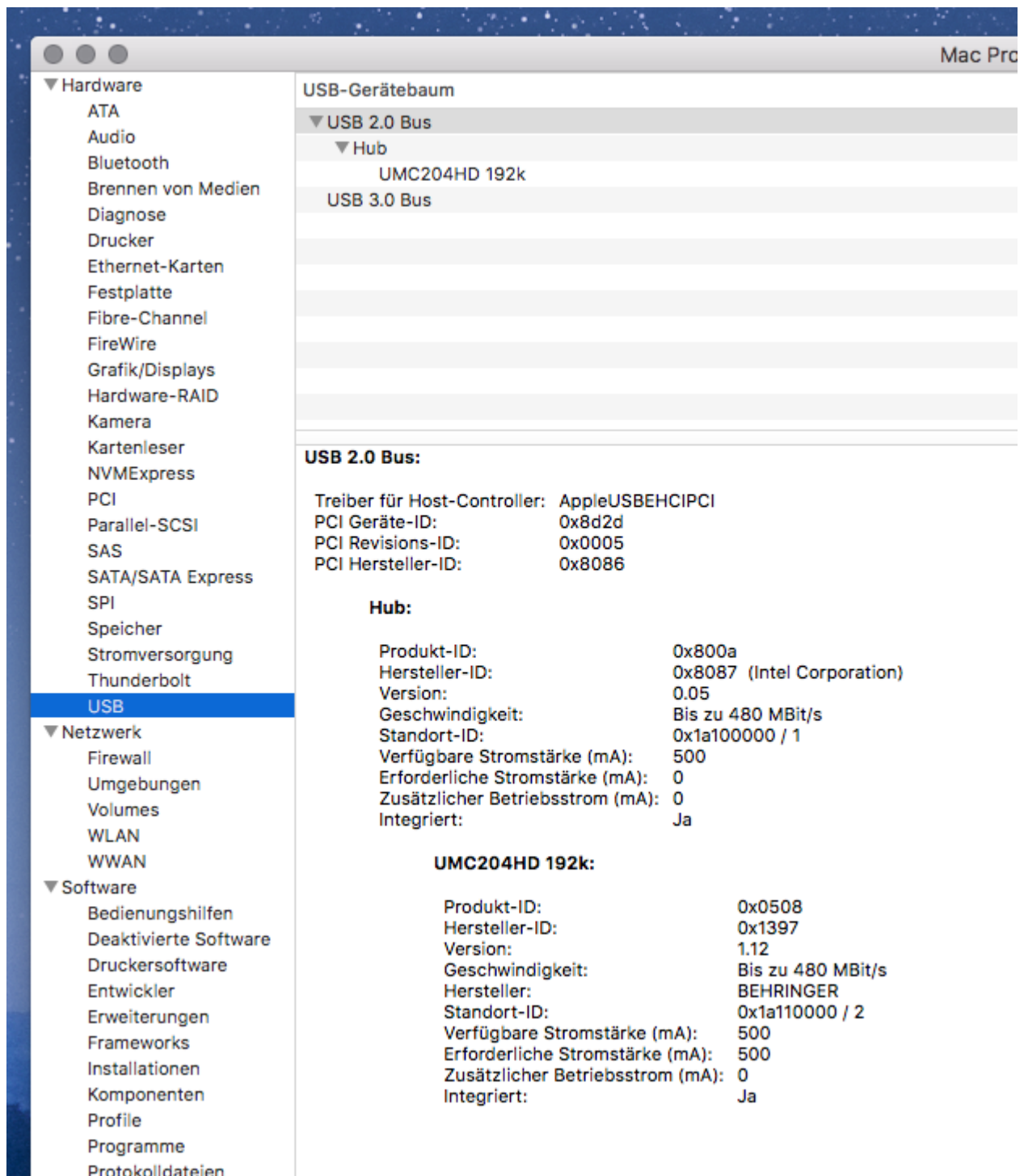
Aus meiner Sicht müssten korrekterweise die USB2 Ports, bzw. USB2 Devices and USB3 Ports zum AppleUSBEHCI geroutet werden. Der Kext FakePCIID_XHCIMux.kext von RehabMan sollte nach der Beschreibung auf [GitHub](#) diese Funktion übernehmen. Er schreibt dazu folgendes:

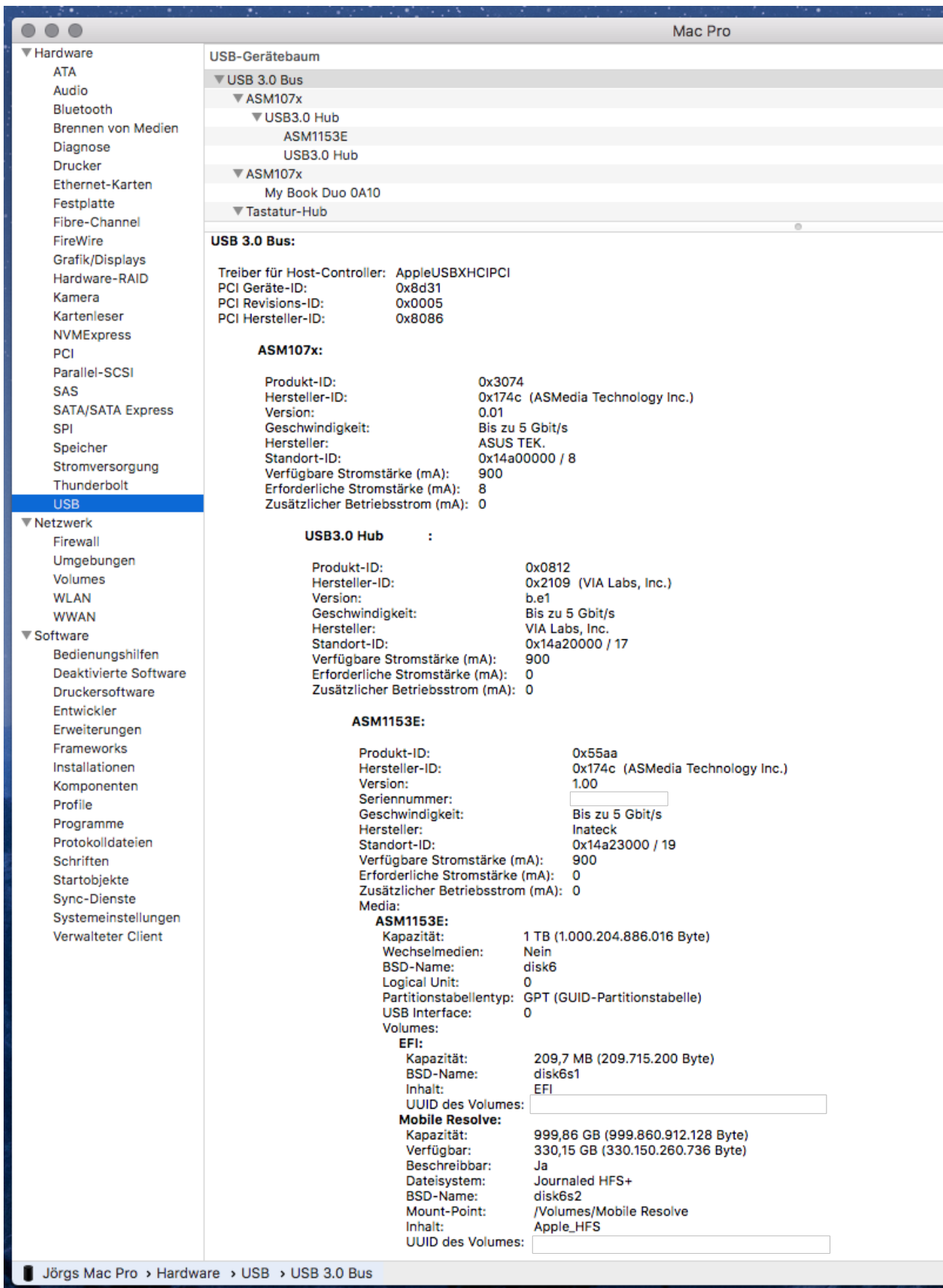
"... The effect is to route any USB2 devices attached to the USB2 pins on the XHC ports to EHCI. In other words, handle USB2 devices with the USB2 drivers instead of the USB3 drivers (AppleUSBEHCI vs. AppleUSBXHCI)."

Aber es hakt im Zusammenspiel und die allgemeine Dokumentationslage ist doch recht dünn. Mit dem Kext sind bis auf einen plötzlich alle Ports weg (erstes Bild) aber der Treiber stimmt -> ohne den Kext sind alle Ports da, aber mit dem bekannten Standart, das alle Ports via AppleUSBXHCIPCI verwaltet werden (zweites Bild).

Kennt sich jemand genauer damit aus und kann mit etwas Hilfestellung leisten?

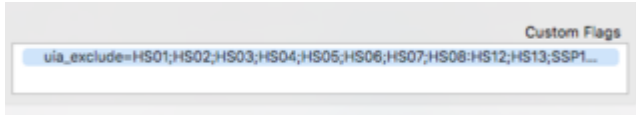
Gruß Joe





Beitrag von „rubenszy“ vom 22. Januar 2017, 21:01

Da du X99 benutzt geht nur USBInjectAll.kext mit [XHCI-x99-injector.kext](#) und in der clover config die USB Anschlüsse abschalten die nicht physikalisch vorhanden sind.



Beitrag von „JoeHidden“ vom 22. Januar 2017, 21:46

Hey [@rubenszy](#)

Vielleicht bin ich gerade ganz auf dem falschen Pfad, aber der Kext beschränkt sich doch auf den AppleUSBXHCI Driver und genau den möchte ich für die USB2 Ports eben nicht verwenden. Oder stehe ich gerade auf dem Schlauch?

Code

1. <dict>
2. <key>CFBundleIdentifier</key>
3. <string>com.apple.driver.usb.AppleUSBXHCIPCI</string>
4. <key>alternate:IOClass</key>
5. <string>AppleUSBXHCILPTH</string>
6. <key>IOClass</key>
7. <string>AppleUSBXHCILPTH</string>
8. <key>IOPCIPrimaryMatch</key>
9. <string>0x8d318086</string>
10. <key>IOPCITunnelCompatible</key>
11. <true/>
12. <key>IOProbeScore</key>
13. <integer>1000</integer>

14. <key>IOProviderClass</key>
15. <string>IOPCIDevice</string>
16. </dict>

Alles anzeigen

Gruß Joe

Beitrag von „rubenszy“ vom 22. Januar 2017, 22:33

USB 2.0 läuft immer was meist erst aktiviert werden muss ist USB 3.0, was du mit diesem kext auch machst.

Wichtig ist noch der DSDT Patch XHC1 to XHC

Beitrag von „kuckkuck“ vom 22. Januar 2017, 23:13

Es geht hier aber nicht um USB 2.0 Funktionalität sondern um den Controller über welchen diese Ports geroutet werden. Dabei ist EHCI häufig zuverlässiger als XHCI weshalb Joe's Idee hier ein guter Ansatz ist.

[@JoeHidden](#) Wichtig ist dass jeglicher anderer USB Patch entfernt ist. Dann brauchst du FakePCIID_XHCIMux und natürlich den FakePCIID, beide funktionieren zuverlässig aus S/L/E. Ein Port Limit Patch darf nicht aktiviert sein und ein anderer Kext wie zb USBInjectAll oder GenericUSBXHCI nicht installiert/aktiviert sein! In der DSDT sind folgende Patches nötig:

- EHC1 --> EH01
- EHC2 --> EH02
- XHC1 --> XHC

Ich bitte dich auch einmal hier vorbei zu schauen und deine [BIOS Settings](#) zu checken: <https://www.hackintosh-forum.d...6-USB-Port-L%C3%B6sungen/>

Falls FakePCIID_XHCIMux bei dir wirklich nicht zuverlässig funtkioniert, können wir die Windows Versions Prüfung in der DSDT modifizieren um die richtige Controller Zuweisung zu bekommen. Dazu aber mehr wenn es wirklich so weit ist.

Beitrag von „JoeHidden“ vom 22. Januar 2017, 23:13

Ich freue mich wirklich das Du Dir die Zeit nimmst und versuchst zu helfen @rubenszy. Das meine ich absolut ernst!

Aer ich muss die Frage stellen: hast Du mein Text gelesen? Auch wenn Du meinst, das USB2 immer geht, habe ich genau hier das Problem.

Sowohl Deinen ersten Tipp uia_exclude, als auch den zweiten Hinweis DSDT-Patch XHC hatte ich doch bereits in meinem ersten Satz geschrieben.

[Zitat von JoeHidden](#)

Zunächst zum IST Stand:

Ich habe via DSDT die Anpassung von XHCI->XHC, EHC1->EH01 sowie ECH2-EH02 durchgeführt. Die neuste Version der USBInjectAll.kext kommt zum Einsatz und alle Devices werden von AppleUSBXHCIPCI bearbeitet. Die uia_exclude sind entsprechend gesetzt, es sind unter 15 Ports und es ist kein Patch der USB Treiber aktiv. Sowohl USB2 als auch USB3 Geräte werden korrekt erkannt und die Geschwindigkeit von USB3 ist korrekt. Im Prinzip entspricht alles der aktuellen [Anleitung](#) von [@kuckkuck](#).

Mittlerweile bin ich soweit, das ich mit dem FakePCIID_XHCIMux.kext zumindest 2 Ports als reine USB2 Ports am laufen habe und parallel zwei Ports als reine USB3 Ports. Das ist schon mal ein Schritt in die richtige Richtung aber das Handover von XHCI zu EHCI (also USB2 Device im USB3 Port) geht leider nicht. Mal sehen...

Beitrag von „kuckkuck“ vom 22. Januar 2017, 23:16

[Zitat von JoeHidden](#)

XHCI

XHCI und XHC1 ist aber von dem Namen in der DSDT her ein Unterschied. Gib außerdem mal bescheid ob in deiner DSDT XHC oder XHC_ geschrieben steht.

Beitrag von „JoeHidden“ vom 22. Januar 2017, 23:16

[Zitat von kuckkuck](#)

Kext wie zb USBInjectAll oder GenericUSBXHCI nicht installiert/aktiviert sein!

Ah, ok. Ich hatte es so verstanden, das der USBInjektALL trotzdem notwendig wäre. Ok, ich probiere das mal [@Kuckkuck](#). Danke!

Nachtrag: Genau so habe ich es in der DSDT angepasst.

EHC1 --> EH01

EHC2 --> EH02

XHC1 --> XHC

Beitrag von „kuckkuck“ vom 22. Januar 2017, 23:18

Nein, hier wird das Problem sein. USBInjectAll macht eher das Gegenteil verglichen mit FakePCIID_XHCIMux. Unbedingt entfernen und die uia_excludelist ist für FakePCIID auch nicht nötig. Und bitte keine Vollzitate 😊👍

Edit: Immernoch die gleiche Frage:

[Zitat von kuckkuck](#)

Gib außerdem mal bescheid ob in deiner DSDT XHC oder XHC_ geschrieben steht.

Beitrag von „JoeHidden“ vom 22. Januar 2017, 23:27

[@Kuckkuck](#) XHC steht in der DSDT

Beitrag von „kuckkuck“ vom 22. Januar 2017, 23:38

ACPI Namen haben immer 4 Buchstaben. Deshalb muss es auch XHC_ heißen. Bei einer config.plist in Plist Editor wäre das dann (in HEX) 5848 435f oder eben WEhDXw== im Text Editor.

Für den Plist editor also Hex benutzen, wenn du faul bist machs so im Terminal:

Code

1. echo -n XHC_|xxd
 2. 0000000: 5848 435f XHC_
-

Beitrag von „rubenszy“ vom 22. Januar 2017, 23:42

Clover DSDT Patch XHCI to XHC

Comment	Find	Replace
XHCI to XHC	58484349	5848435F

Beitrag von „kuckkuck“ vom 22. Januar 2017, 23:43

Genau, dort steht es ja auch mit XHC_ geschrieben 😊

Beitrag von „al6042“ vom 22. Januar 2017, 23:44

[@kuckkuck](#)

Die Info mit den 4 Buchstaben irritiert mich auch ein wenig.

Jedenfalls wenn es um Devices geht.

Immerhin gibt es auch den MacIASL Patch "USB - Device Renames for OSX 10.11" der folgendes beinhaltet:

Code

1. into device label EHC1 set_label begin EH01 end;
 2. into device label EHC2 set_label begin EH02 end;
 3. into device label XHC1 set_label begin XHC end;
 4. into_all all code_regex EHC1 replaceall_matched begin EH01 end;
 5. into_all all code_regex EHC2 replaceall_matched begin EH02 end;
 6. into_all all code_regex XHC1 replaceall_matched begin XHC end;
-

Beitrag von „JoeHidden“ vom 22. Januar 2017, 23:49

Ok, hab ich angepasst.

Wenn ich nur mit der FakePCIID_XHCIMux & FakePCIID boote ist USB2 ok. USB3 fehlt.

Zu den [BIOS Einstellungen](#). Die zwei Handover Values sind so wie verlinkt. Der XHCI steht aber auf Smart Auto, da bei Enabled das BIOS vom Board die EHCI Controller deaktiviert - was ja das Ziel untergeben würde... 😊

Dementsprechende führt der Wechsel auf XHCI Enable im Bios zum umgekehrten Bild. USB3 ok
- USB2 fehlt.

Gruß Joe

Beitrag von „kuckkuck“ vom 23. Januar 2017, 15:35

[@al6042](#) Im ACPI Spec steht das alles genauestens beschrieben, jedoch auf Kosten der Übersichtlichkeit. Nach meinem Wissen sollen ACPI Namen, Knoten und Vendor IDs 4 Buchstaben (A-Z großgeschrieben) oder _ besitzen. Wie sich das benutzen von nur 3 Ziffern auswirkt kann ich nicht genau sagen. Strengstens nach ACPI Spec müsste es aber XHCI/XHC1 --> XHC_ heißen, um den Grundsätzen zu folgen. ACPI Spec gibts im Internet, hier ein paar Regeln zu [Knoten](#) und zu [ACPI IDs](#).

[@JoeHidden](#), gibt es bei dir auch die Möglichkeit "Auto" im Bios?

Beitrag von „JoeHidden“ vom 23. Januar 2017, 16:54

[@kuckkuck](#) Ja, Auto geht auch. Macht zumindest äußerlich erst mal keinen Unterschied.

Kurz zum aktuellen Stand. Ich habe gestern noch etwas geforscht.

- Ich habe noch einmal validiert das die Adresse XHCI Controller im FakePCIID_XHCIMux 0x8d31/0x8086 korrekt vorhanden ist. Das trifft zu.
- Von den zwei EHCI Controllern werden auch zwei erkannt, allerdings nur einer mit Anschlüssen. Da muss ich aber nochmal genau die Zuordnung vom Board studieren.
- Nur mit der FakePCIID_XHCIMux & FakePCIID boote ist entweder USB2 oder USB3 vorhanden / je nach XHCI Bios Setting.

Für den produktiven Betrieb habe ich deshalb zusätzlich nochmal die USBInjektAll (nur

temporär Kuckuck - nicht schlagen) aktiviert. In Kombination der beiden Kexte habe ich einen USB2 Hub und einen USB3 Hub im System und mittels Hardware Hubs bekomme ich alle Geräte auf den korrekten Controllern ans rennen.

Positiv ist, das in der Konstellation das USB Audio Interface nun perfekt via EHCI funktioniert-der Weg ist also der richtige... 😊

Negativ ist, das die USB3 Ports nicht abwärtskompatibel sind, was für den Moment erstmal ok ist.

Ich versuche nachher mal die einzelnen Adressen der Controller und das Hardwarelayout nachzusehen. Dann sehen wir weiter... 😊

Gruß
Joe

Beitrag von „kuckkuck“ vom 23. Januar 2017, 17:21

Dann wünsche ich dir erstmal noch erfolgreiches Testen 👍

Ansonsten empfehle ich dir beim jetzigen Stand die Windows Versions Prüfung bei Darwin in der DSDT zu modifizieren. Dadurch sollten wir Abwärtskompatibilität und USB2 auf EHCI bei deinem Hackintosh einrichten können. Guide dazu habe ich hier im entsprechenden Spoiler verfasst: [USB Port Lösungen/FAQ](#)

Beitrag von „JoeHidden“ vom 23. Januar 2017, 17:22

Ok, das sehe ich mir an. Danke nochmal!

Beitrag von „kuckkuck“ vom 23. Januar 2017, 19:13

Aber bitte immer schön die DSDT backuen, die Methode ist nicht gerade die am weitesten verbreitetste Methode. Wir müssen geg. ein wenig mit _OSI Simulationen rumspielen um eine passende Windows Version zu finden, die für dein System passt. Außerdem brauchst du für diese Methode kein FakePCIID_XHCIMux oder anderen USB Kext! Alle vorgänge laufen direkt durch das ACPI.

Beitrag von „JoeHidden“ vom 23. Januar 2017, 20:40

Klar immer nur mit Backup [@kuckkuck...](#) 😊

Ich bin gerade mit der XOSI Variante dran. Ich kann sehen (wenn ich mit MaciASL die aktuelle Tabelle auslese, das die XOSI Methode in der DSST vorhanden ist. Bisher bemerke ich im System keine Veränderung. Gibt es eine Stelle wo ich validieren kann ob die Einstellungen überhaupt zu einer Veränderung führen? Fühlt sich gerade etwas "blind" an.

Ich nehme an, das die Umbenennungen XHCI->XHC, EHCI->EH01, etc. auf bei der Methode aktiv bleiben oder habe ich das missverstanden?

Nachtrag: Im MaciASL sehe ich auch die SSDT-XOSI im ACPI.

Code

1. DefinitionBlock ("iASLapumY7.aml", "SSDT", 2, "hack", "XOSI", 0x00000000)
2. {
3. Method (XOSI, 1, NotSerialized)
4. {
5. Store (Package (0x07)
6. {
7. "Windows",
8. "Windows 2001",
9. "Windows 2001 SP2",
10. "Windows 2006",
11. "Windows 2006 SP1",

12. "Windows 2009",
13. "Windows 2012"
14. }, Local0)
15. Return (LNotEqual (Ones, Match (Local0, MEQ, Arg0, MTR, Zero,
16. Zero)))
17. }
18. }

Alles anzeigen

Beitrag von „kuckkuck“ vom 24. Januar 2017, 00:09

Ja das ist alles etwas "blind" und merken kannst du nur dann wenn sich etwas mit deinen USB Ports etwas verändert... Wir müssen experimentieren.

Dein SSDT Auszug ist aus der AML Datei welche Windows 8 simulieren würde. Wieso das so ist kann ich dir später erklären. Jetzt erstmal: Wo hast du diese aml gespeichert? Wo ist die DSL Datei und unter welchem Namen hast du die beiden jeweils gespeichert? Ist die SSDT.aml in der config.plist in der sorted order? Benutzt du eine SSDT für Speedstep?

Beitrag von „JoeHidden“ vom 24. Januar 2017, 11:34

[Zitat von kuckkuck](#)

Wir müssen experimentieren.

Ok, ist ja auch mal spannend.

[Zitat von kuckkuck](#)

Wo hast du diese aml gespeichert? Wo ist die DSL Datei und unter welchem Namen

hast du die beiden jeweils gespeichert?

EFI/CLOVER/ACPI/patched/SSDT-XOSI.aml
_OSI zu XOSI zunächst via Clover Patch (ACPI-->DSDT-->Patches)

[Zitat von kuckkuck](#)

Ist die SSDT.aml in der config.plist in der sorted order?

Ja

[Zitat von kuckkuck](#)

Benutzt du eine SSDT für Speedstep?

Nein

[Zitat von kuckkuck](#)

Wieso das so ist kann ich dir später erklären.

Ich denke das habe ich schon verstanden. Windows 8 weil sowohl für Windows 8 (_OSI string: "Windows 2012") und alle früheren Windows Versionen ein 0xFFFFFFFF als Return value liefern. Soll Windows7 gemeldet werden, müsste der Eintrag "Windows 2012" entfernt werden. Damit wäre "Windows 2009" respektive Windows 7 der höchstwertige valide Eintrag. Korrekt?

Beitrag von „kuckkuck“ vom 24. Januar 2017, 15:19

Genau so ist es. Da Windows 8 nicht will, probieren wir jetzt erst einmal Windows 7. Dazu einfach in der DSDT.dsl Windows 2012 als Bemerkung/Kommentar setzen (//) und danach neu als .aml Compilieren und einbinden. Dadurch werden alle Bemerkungen entfernt und somit auch Windows 8, was dann zum aktivieren von _OSI Windows 2009 führt. _OSI Windows und

alle vorherigen Versionen müssen -->true-return liefern und dürfen deshalb nicht entfernt werden. Die Server Versionen aka x.1 lassen wir auch raus 😊
Wenn du keine SSDT für Speedstep benutzt, benenn die SSDT-XOSI.aml mal in SSDT.aml um 👍

Beitrag von „JoeHidden“ vom 24. Januar 2017, 17:48

Hallo [@Kuckuck](#)

ich habe nochmal beide Varianten ohne jegliche Injektoren etc. getestet. Weder Windows 7 noch Windows 8 scheinen eine Veränderung zu ergeben. Über den Weg MaciASL/New from AHCI/SSDT kann man sehen, das die Änderungen jeweils übernommen wurden, mehr aber auch nicht. Vielleicht verhält sich das ASUS Bios in beiden (allen) Fällen gleich?

Ohne Injektor funktionieren alle USB2 Ports übrigens auch perfekt. Mit Rehabman's XHCI-x99-injector.kext wird alles gnadenlos über den AppleUSBXHCIPCI geroutet was logischerweise dazu führt das auch die USB3 Geräte laufen, aber damit auch die Audio Probleme vorhanden sind. Was mich wunder ist, das der XHCI-x99-injector eigentlich nur 0x8d318086 auf den AppleUSBXHCIPCI bindet. Warum wirkt sich das auf die USB2 Devices am Controller 0x8d2d8086 aus? Ich meine es wäre ja ok, wenn die USB2 Devices die in einem USB3 Port stecken auch vom xHCI verwurstelt werden, aber warum kassiert der auch alle USB2 Ports vom EHCI ein?

Gruß Joe

Beitrag von „kuckuck“ vom 24. Januar 2017, 18:27

Gut dann machen wir das eben radikaler...

Öffne mal ein Backup von einer DSDT, die noch keine Veränderungen bezüglich OSI oder EHCI-->EH01... hat. Dann hol dir Rehabmans Laptop Repo für MaciASL und apply den OS Check Fix für Windows 7. Dabei wir die _OSI Darwin Methode jetzt komplett ersetzt. Mal sehen

was passiert. Windows 8 und als letztes Vista kannst du natürlich auch probieren. Und wenn das auch nicht will gibt es eine noch etwas radikalere Lösung...

Zu deinen Versuchen mit FakePCIID_XHCIMux: hattest du nach installieren des Kexts mal mit den XHCI Modes im BIOS gespielt?

Beitrag von „JoeHidden“ vom 24. Januar 2017, 19:28

Radikaler getestet - ohne Befund 😊

Ich sehe in der ausgelesenen DSST den Eintrag

Code

1. If (LOr (_OSI ("Darwin"), _OSI ("Windows 2009")))
2. {
3. Store (0x0E, OSVR)
4. }

Und die Controller heißen wieder EHC1, EHC2 und XHCI - insofern wurde alles angewendet. Einen Unterschied im Verhalten kann ich nicht feststellen. Ich werde jetzt mal den Patch für Windows 8 Testen und ggf. mal Windows 10.

Nachtrag: Windows 8 bringt auch keine Veränderung. Windows 10 entfällt alt Test, da in der ausgelesenen DSDT kein Eintrag zu "Windows 2015", was laut MS Doku Windows 10 wäre, existiert.

Beitrag von „kuckkuck“ vom 25. Januar 2017, 19:27

Was ist mit Vista, was ist mit meiner letzten Frage aus der Nachricht oben?

Beitrag von „JoeHidden“ vom 25. Januar 2017, 19:35

Entschuldige, hatte ich nicht explizit erwähnt. Identisch.

Beitrag von „kuckkuck“ vom 25. Januar 2017, 19:49

[Zitat von kuckkuck](#)

Zu deinen Versuchen mit FakePCIID_XHCIMux: hattest du nach installieren des Kexts mal mit den XHCI Modes im BIOS gespielt

Beitrag von „JoeHidden“ vom 25. Januar 2017, 20:04

Verdammt 😊 Ja, hab ich auch. Gefühlt läuft nur Smart Auto wirklich gut. Bei Auto treten USB3 Disconnects auf.

Am besten scheint im Moment die folgenden Variante zu sein.

DSDT: Nur EHC1/EH01, EHC2/EH02, XHCI/XHC_

Kext: FakePCIID+FakePCIID_XHCIMux+USB_Injector x99

Clover Boot Argument -uia_exclude_hs

Bisher laufen alle Ports perfekt stabil. Und durch das Boot Argument sind auch nur die SSP Ports auf den XHCI gebunden.

Die Abwärtskompatibilität der USB3 Ports geht aber immer noch nicht.

Ich frage mich langsam ob ich nicht einen Denkfehler habe und da in der Konstellation

überhaupt nicht möglich ist?

Kann es sein, das der FakePCIID_XHCIMux per Definition die Abwärtskompatibilität an den USB3 Ports kill?

Und wenn nicht, kann es daran liegen das bei meinem System EH01 in der IORegistry komplett leer sind und alle Ports am EH02 hängen? Vielleicht ein Zielkonflikt?

Gruß Joe

Beitrag von „kuckkuck“ vom 25. Januar 2017, 20:36

Hmm... FakePCIID_XHCIMux killt nicht die Abwärtskompatibilität, er routet nur Ports zu einem EHC Controller...

[Zitat von JoeHidden](#)

Und durch das Boot Argument sind auch nur die SSP Ports auf den XHCI gebunden.

Das Boot Argument tut absolut garnichts ohne USBInjectAll.kext. Das wird nichts 😊
Kurze Frage, wie verhalten sich deine USB Ports bezüglich des Mikros wenn kein einziger Patch bezüglich USB realisiert ist?

Radikalitäts-Steigerung + 1 👍 : Such dir mal die ESEL Methode in der DSDT raus

Und jetzt killen wir die einfach mal... 😄

Code

1. Method (ESEL, ...)
 2. { Return(0)
 3. ...
 4. }
-

Beitrag von „JoeHidden“ vom 25. Januar 2017, 21:28

[Zitat von kuckkuck](#)

Hmm... FakePCIID_XHCIMux killt nicht die Abwärtskompatibilität, er routet nur Ports zu einem EHC Controller.

Ok, dann ist die Frage ob er das in irgendwie mit den ACPI Namen korrespondiert. Vielleicht ist da EH01 hardcoded drin, oder sowas? Mich wundert nur das bei mir alles am EH02 hängt und der EH01 leer ist.

[Zitat von kuckkuck](#)

Das Boot Argument tut absolut garnichts ohne USBInjectAll.kext. Das wird nichts 😊

Nein, das ist nicht korrekt. Ohne das Argument werden die HSxx Ports an den XHC gebunden - mit nicht. Ich nehme an das der FakePCIID_XHCIMux den Parameter auch auswertet. Der Hinweis zum Einsatz kam von Rehabman in einem anderen Thread den ich dazu gelesen hatte.

[Zitat von kuckkuck](#)

Kurze Frage, wie verhalten sich deine USB Ports bezüglich des Mikros wenn kein einziger Patch bezüglich USB realisiert ist?

Mikro beziehst Du auf das USB Audio Device? Wenn ja, ist die Antwort - läuft korrekt ohne jeden Kext als USB2 Device. Ohne Kexte fehlt dann aber der USB3 Support.

[Zitat von kuckkuck](#)

Radikalitäts-Steigerung + 1 👍

Ich werde testen und berichten... 😊

Beitrag von „kuckkuck“ vom 25. Januar 2017, 22:10

[Zitat von JoeHidden](#)

Der Hinweis zum Einsatz kam von Rehabman in einem anderen Thread den ich dazu gelesen hatte.

Das will ich sehen... Glaube ich nicht, aber vielleicht hast du recht...
Wie wirkt sich das Nutzen des Bootargs mit FakePCIID_XHCIMux auf den XHC Controller in IOReg aus? Was ist zu sehen?

[Zitat von JoeHidden](#)

Ich werde testen und berichten...

Dann tu das mal, ich bin gespannt was bei deinem System passiert...

Wenn das nicht funktioniert (und auch nicht die deaktivierung von XSEL (machen wir später)) dann rate ich dir dazu mal den Port Limit Patch zu probieren. Wie das geht ist auch in der FAQ beschrieben. Das macht vielleicht insofern Sinn, als das deine USB Ports ohne Patch gut zu funktionieren zu scheinen. Ich bin gespannt.

Beitrag von „JoeHidden“ vom 25. Januar 2017, 22:41

[Zitat von kuckkuck](#)

Das will ich sehen... Glaube ich nicht, aber vielleicht hast du recht...

Läuft... 😊 Jetzt wollte ich Dir gerade den Quote reinsetzen weil ich den Tab noch offen hatte, da seh ich das es die Antwort auf einen Post von der Seite davor war und der hatte natürlich doch die USBInjektAll. Mea Culpa 😊

Aber ich bin auch bei Rehabman auf die Aussage gestolpert: "By default forces all XHCI ports to route USB2 devices to **EHC1**."

Ich probier noch mal eben was und dann teste ich weiter... 😊

Beitrag von „kuckkuck“ vom 26. Januar 2017, 14:48

Das mit EHC1 oder 2 ist normal und kommt aufs SMBios drauf an 😊 Welche System Definition benutzt du denn?
Wie sind deine Tests ausgegangen?

Beitrag von „JoeHidden“ vom 26. Januar 2017, 14:54

MacPro 6,1

Ich hab gestern Abend früh Schluss gemacht und teste heute Abend weiter.

[@Kuckkuck](#) Codeword "Der ESEL ist offline" 😊 Leider auch keine signifikanten Änderungen erkennbar.

In der Info.plist der FakePCIID_XHCIMux finden sich die RM,pr- Einträge.

▼ XHCIMux 8 and 9-series	Dictionary	↕ 7 Schlüssel/Wert-Paare
CFBundleIdentifier	String	↕ org.rehabman.driver.FakePCIID.XHCIMux
▼ FakeProperties	Dictionary	↕ 6 Schlüssel/Wert-Paare
RM,pr2-block	Daten	↕ 1 Bytes: 00
RM,pr2-chipset-mask	Daten	↕ 4 Bytes: FF3F0000
RM,pr2-force	Daten	↕ 4 Bytes: 00000000
RM,pr2-honor-pr2m	Daten	↕ 1 Bytes: 01
RM,pr2-init	Daten	↕ 1 Bytes: 01
RM,pr2m-block	Daten	↕ 1 Bytes: 01
IOClass	String	↕ FakePCIID_XHCIMux
IOMatchCategory	String	↕ FakePCIID_XHCIMux
IOPCIPrimaryMatch	String	↕ 0x9c318086 0x9cb18086 0x8c318086 0x8cb18086 0x8d318086
IOProbeScore	Zahl	↕ 9.001
IOProviderClass	String	↕ IOPCIDevice

Wirklich viel habe ich dazu nicht gefunden aber Rehabman schreibt dazu folgendes:

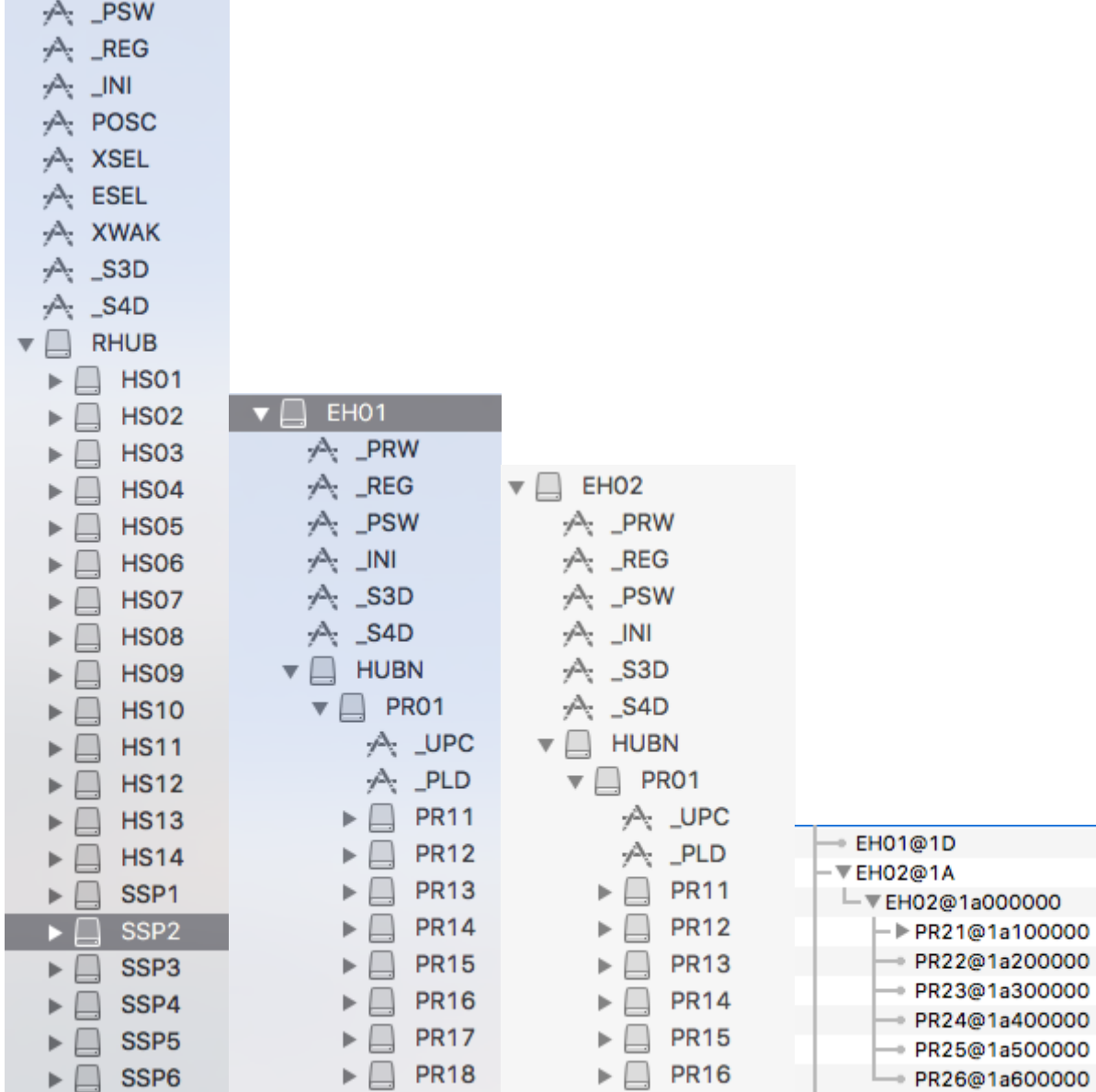
Zitat

Configuration properties and their defaults: RM,pr2-force <00 00 00 00>. By default forces all XHCI ports to route USB2 devices to EHC1. RM,pr2-init <01>. Will write RM,pr2-force value at startup if non-zero. RM,pr2-block <01>. Will block writes to XUSB2PR if non-zero. RM,pr2m-block <01>. No evidence that OS X drivers attempt to write XUSB2PRM (offset 0xD4), but since this next relies on a valid value here (as provided by the BIOS), writes to it are blocked if non-zero. RM,pr2-honor-pr2m <01>: Changes to XUSB2PR will be masked by XUSB2PRM if this is non-zero. RM,pr2-chipset-mask: Writes to XUSB2PR are masked by this value. This is defined by the chipset documentation. Default value depends on chipset. Refer to Intel 7/8/9-series chipset data sheet for more info.

So wie ich das verstehe wäre es doch damit möglich per RM,pr2-force nur die reinen USB 2.0 Ports an den EHCI zu leiten, oder? Das würde ja schon ausreichen. Ein USB 2.0 Device am USB 3.0 (Dongles etc.) dürfte gern auch XHC bearbeitet werden. Bisher erschließt sich mir die Maskierung aber überhaupt nicht - bzw. die Nummerierung auf deren Basis dann maskiert wird. Wirst Du da schlau draus?

--- Themensprung / Warnung 😊 ---

Auf die Gefahr Dir jetzt vollkommen auf die Nerven zu gehen... aber ich möchte das wirklich verstehen. Ich habe mir die DSDT nochmal sehr genau angesehen. Speziell das Thema korrekte _UPC Objekte. Beim XHC stimmen die Angaben von DSDT und IOReg überein, also Portbezeichnungen / Anzahl / Adressen. Bei EH01 und EH02 nicht. Bei EH01 (Adresse 8d26) fehlt der HUB und es fehlen alle Ports, selbst mit USBInjektAll ohne Filter. Bei EH02 werden die sechs Ports korrekt übernommen, aber die Nummern der Ports werden um zehn inkrementiert? Es taucht dem entsprechend auch nur ein USB2.0 Bus im USB Gerätebaum auf. Ich finde das nach wie vor sehr seltsam.



Beitrag von „kuckkuck“ vom 27. Januar 2017, 11:07

Zum Anfang:

Das ganze ist von mehreren dingen abhängig. RM,pr2-force routet zwar zu EHC, wird aber nur aktiviert wenn RM,pr2-init aktiviert ist um beim Start RM,pr2-force zu aktivieren. XUSB2PR wird geblockt damit bei dem ganzen Prozess Apples Treiber nicht zwischen funken. Lediglich das

Chipset Value könnte nötige Veränderungen bringen, was aber überhaupt garnicht dokumentiert ist... Insofern ist das ganze eher eine Sackgasse für uns 😞

Das mit der DSDT mag komisch sein, aber ich kann dir nicht sagen ob das Behavior wirklich falsch ist oder sogar so stimmt... Ich mein der eine Controller wird weggelassen, der andere in teilen der Bezeichnungen inkrementiert. Wer weiß. 😄

Ich weiß nur was wir als nächstes versuchen könnten, und das wäre XSEL statt ESEL zu deaktivieren. Gleiche Prozedur wie mit ESEL. Wir wollen ja Abwärtskompatibilität erschaffen, nicht aufwärts 😄

Kurze Frage nebenbei, wie weit ist deine Standard DSDT gepatcht? Sind da wenigstens die basics drin?

Beitrag von „JoeHidden“ vom 27. Januar 2017, 12:18

Ok, dann gehe ich nachher mal XSEL an. Danke für die Erklärung!

Die DSDT umfasst derzeit, die XHCI/EHC1/EHC2 Änderung, die Anpassung der GFX für die PCIID meiner 980ti (MacPro6,1). Dazu die Basics aus dem Wiki, allerdings ohne HDMI und Audio Einträge. Ich nutze den internen ALC Audio Chip nicht. OS Check Fix und USB Power auch nicht.

Gruß Joe

[@Kuckuck](#) - Auch mit XSEL keine Änderung.

Beitrag von „kuckkuck“ vom 28. Januar 2017, 06:18

Und hast du mal den port limit patch probiert?

Beitrag von „JoeHidden“ vom 28. Januar 2017, 10:08

Bringt der in dem Kontext denn noch etwas. Ich habe den schon seit einer weile raus, weil ich ja die passenden Port ausgewählt habe und exkludiere was nicht gebraucht ist auf dem XHCI?

Ich bin aber an anderer Front weiter. Rehabman meint, das Problem liegt an fehlenden Injects bei EHCI. Ich denke ich muss herausfinden, warum die EH01/EH02 Ports nicht korrekt vorhanden sind.

Beitrag von „kuckkuck“ vom 29. Januar 2017, 08:32

Was mich mal interessieren würde, wäre wie das ganze unter Windows aussieht. Wenn es dort ebenfalls ähnlichem Probleme gibt könnte es am Mainboard oder dem [BIOS settings](#) liegen. Probleme unter Windows bezweifle ich jedoch....

Halt mich auf dem laufenden was rehabman dir vorschlägt 😊

Beitrag von „JoeHidden“ vom 29. Januar 2017, 12:46

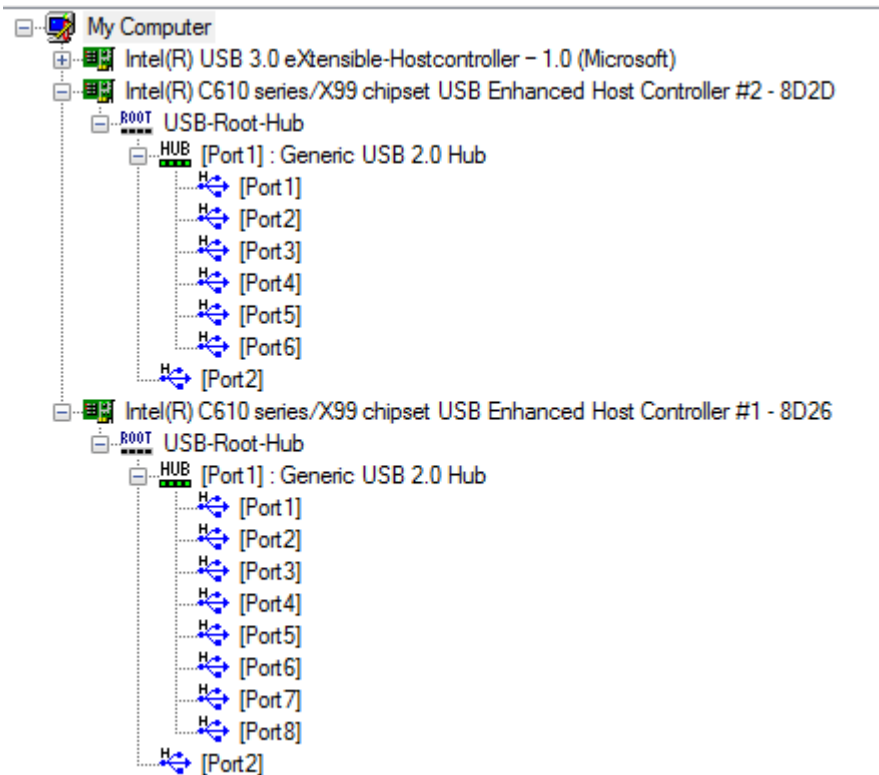
Unter Windows 10 ist alles wie es sein soll. Bisher fischen wir im trüben... 😊 Er meinte der Clover Parameter "Devices/USB/FixOwnership=true" wäre entscheidend. Ich hatte den Wert auf false, da im Clover Wiki dazu steht "This fix is not relevant for UEFI booting.". Er meinte dazu, die Angabe im Clover Wiki stimmt nicht. Einen Unterschied macht es leider trotzdem nicht. Die Logs helfen aber auch nur bedingt weiter. Interessant finde ich die folgenden Zeile aus dem Kernellog.

Code

1. kernel: EH02: match category IODefaultMatchCategory exists

Die taucht bezüglich EH01 nicht auf. Ich habe aber auch google etc. schon mit allen sinnvollen

Suchkombis durch. Es gab durchaus andere Fälle, denen auf dem X99 der EH01 fehlte - aber die waren dann immer damit fertig wenn sie USB3 am rennen hatten und dann spielt das Problem ja auch keine Rolle mehr... 😊 Naja, ich habe mit meinem Hacky soviel von der Community profitiert, da kann ich jetzt auch mal was erforschen... 😊 Und ich habe durch die systematische Testerei soviel mehr gelernt, als mit "mach mal den USBInjectALL Kext rein, dann gehts". Ist schon ein nettes Hobby.



Beitrag von „kuckkuck“ vom 29. Januar 2017, 14:03

Wenn wir auf 0 wären, würde ich jetzt vorschlagen win 10 zu simulieren. Bei dir gibt es aber anscheinend ein generelles Problem mit EH01 und da ist mein Wissen wirklich zuende. Auch ich habe durch unsere Tests so einiges gelernt!

Zur Zukunft: Sogar Rehabman hat keine konkreten Idee obwohl der so ziemlich den kompletten ACPI Spec auswendig kennt 😊 Dementsprechend wünsche ich die viel glück mit deinen Forschungen! Bei kleineren Teilfragen kann ich dir gerne versuchen zu helfen, ich denke mit ACPI Hintergründen kennen sich auch [@al6042](#) [@Thogg Niatiz](#) [@griven](#) und [@Download-Fritz](#)

aus. Vielleicht können wir hier ja irgendwas auf die Beine stellen 😊

Beitrag von „mhaeuser“ vom 29. Januar 2017, 14:18

[Zitat von JoeHidden](#)

"Devices/USB/FixOwnership=true" wäre entscheidend. Ich hatte den Wert auf false, da im Clover Wiki dazu steht "This fix is not relevant for UEFI booting.". Er meinte dazu, die Angabe im Clover Wiki stimmt nicht.

"Fix Ownership" is quasi ein Clover-seitiges "EHCI/XHCI Handoff".

Hatte nicht den Nerv, mir den ganzen Thread durchzulesen, werd' ich aber vielleicht heute Abend machen... mit ACPI kann ich wenig helfen, aber vielleicht fällt mir irgendwas auf. Mal RehabMan angeschrieben?

Beitrag von „JoeHidden“ vom 29. Januar 2017, 14:39

[Zitat von kuckkuck](#)

Wenn wir auf 0 wären, würde ich jetzt vorschlagen win 10 zu simulieren. Bei dir gibt es aber anscheinend ein generelles Problem mit EH01 und da ist mein Wissen wirklich zuende. Auch ich habe durch unsere Tests so einiges gelernt!

Die "höchstmögliche" seitens Asus in der DSDT enthalten Windows Version ist 2013. Die hatten wir bereits getestet. 2015 würde doch dann eh wie 2013 behandelt, oder? Ich frag mich Dia ganz Zeit warum der EH01 in der IOReg vorhanden ist aber der Treiber nicht geladen wird. Wenn er nicht da wäre ok, aber so - seltsam. Auf jeden Fall nochmals Danke bis hierhin [@Kuckkuck](#).

[Zitat von Download-Fritz](#)

"Fix Ownership" is quasi ein Clover-seitiges "EHCI/XHCI Handoff". Hatte nicht den Nerv, mir den ganzen Thread durchzulesen, werd' ich aber vielleicht heute Abend machen... mit ACPI kann ich wenig helfen, aber vielleicht fällt mir irgendwas auf. Mal RehabMan angeschrieben?

Ok, Danke auch Dir schonmal vorab. Mit Rehabman schreibe ich bei IM seit Freitag Abend. Die Kommunikation zieht sich aber wegen der Zeitverschiebung ein wenig... 😊 Letzter Stand war seine Vermutung, das irgendwas im Kernellog stehen müsste. Aber da finden ich nur den oben Zitieren Eintrag zu EH02 (der funktioniert). Zu EH01 schweigt sich der Kernel aus.