

Erledigt

Falls der Eintrag "Deiner Apple Watch das Entsperren deines Mac erlauben" unter Sicherheit / Allgemein fehlt.

Beitrag von „Si Vis Pacem“ vom 20. November 2018, 16:01

1. Mit IOJones nachsehen welche IOClass und welcher Treiber genutzt wird.

// Bei mir hat der Eintrag trotz kompatibler HW gefehlt.

~~2. Dann in der AirportBrcmFixup.kext in der Info.plist alle 4360 Einträge durch NIC ersetzt~~

2. `<string>pci14e4,43b1</string>` aus `<key>Airport_Brcm4360</key>` raus und in `<key>Airport_BrcmNIC</key>` unter IONameMatch rein

3. Nach einem Cache-Update und Reboot ist der Eintrag da.

Hoffe, dass dies Einigen Suchenden hilft.

Nachtrag:

Wußte gar nicht, dass es Continuity/Handoff auch von der Apple Watch zum Mac gibt.

Beitrag von „anonymous_writer“ vom 20. November 2018, 16:20

Dauerhaft kannst dir dazu eine SSDT erstellen so wie hier im Beispiel. Dann musst den Kext beim Update nicht mehr ändern.

[Broadcom bcm94352z DW1560 - Bluetooth 4.0](#)

Beitrag von „Si Vis Pacem“ vom 20. November 2018, 16:47

Danke für den Tipp! Hatte schon fleissig mitgelesen.

Geht leider nicht bei mir.

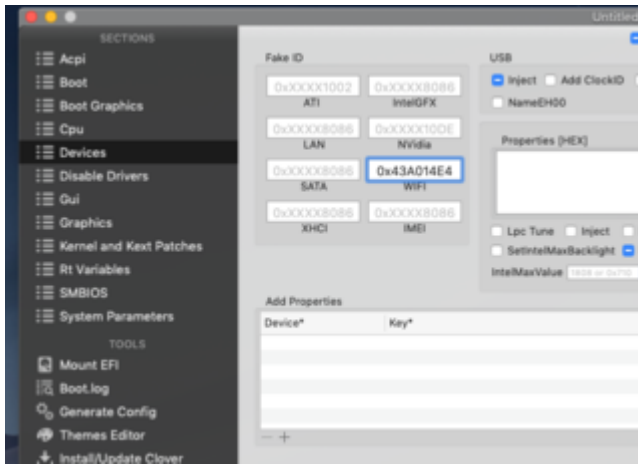
Meine befindet sich unter PEG1, wäre ja an sich kein Problem, wird aber als pci14e4,43b1 gelistet. Also PEG1.pci14e4,43b1.

Und damit lässt sich keine SSDT bauen, da die Syntax zerlegt wird. Zumindest meines Wissenstandes nach.

Unter Clover gibt es sicher die Möglichkeit, aber ich bin leider sehr mau was Clover betrifft. Komme von Oz und bin wegen Mojave gewechselt.

Beitrag von „anonymous_writer“ vom 20. November 2018, 19:09

Über Clover sollte es eigentlich auch gehen. Habe ich aber noch nie getestet. 😊



Wenn du möchtest versuche ich das gerne mal mit der SSDT. Benötige dazu deine DSDT.aml und die IOREG. Die IOREG kannst auch gerne über PM zusenden.

Beitrag von „Si Vis Pacem“ vom 20. November 2018, 19:34

Danke dir!

DSDT nutze ich momentan nicht, da ich es einfach nicht schaffe, mit der DSDT mein DP Audio hinzubekommen.

Heute ist es schon ein bisschen spät und Junior kommt gleich vom Fechten zurück und braucht noch ein warmes Abendessen.

Morgen früh poste ich dann meine SSDTs und meine ioreg.

Schönen Abend!

Beitrag von „anonymous_writer“ vom 20. November 2018, 19:58

Du müsstest dann deine Original DSDT posten.

Beitrag von „Si Vis Pacem“ vom 21. November 2018, 14:40

Spät, aber doch 😊

Beim Dump waren die beiden SSDTs nicht im patched Folder, falls es einen Unterschied macht.

Beitrag von „anonymous_writer“ vom 21. November 2018, 14:49

Verbaut ist eine BROADCOM BCM94352Z DW1560, richtig?

Ich würde gerne was neues Probieren.

Lade dir bitte dieses kleine Tool herunter:

<https://github.com/acidanthera/gfxutil/releases>

Und poste dann bitte zusätzlich die Ausgabe von folgendem Befehl im Terminal:

Code


1. `./gfxutil -f pci14e4,43b1`
-

Beitrag von „Si Vis Pacem“ vom 21. November 2018, 15:19

Azurewave CE-123H

GFXUtil produziert leider ein "DevicePath not found"

Beitrag von „anonymous_writer“ vom 21. November 2018, 15:35

Interessant 

Liefert folgender Befehl eine Ausgabe?

Code

1. `./gfxutil -f pci1a3b,2123`

Bzw: Was zeigt der [DPCIMANAGER](#) in der PCI Liste an?

Beitrag von „Harper Lewis“ vom 21. November 2018, 15:37

[anonymous_writer](#) : Erinnerst du dich noch an das Device THSS? Ein ähnliches Vorgehen sollte doch auch hier per SSDT möglich sein.

Beitrag von „Si Vis Pacem“ vom 21. November 2018, 15:37

Anbei der Screenshot vom DPCIManager.

```
@./gfxutil -f pci1a3b,2123:
```

```
"DevicePath not found"
```

```
@./gfxutil -f pci14e4,43b1
```

Zuerst hat mir gfxutil nur "DevicePath not found" ausgespuckt.

Nachdem ich Windows gebootet hatte und dann zurück in MacOs bin, gibt gfxutil mit dem gleichen Befehl - aus der history geholt - dies aus:

```
"  
DevicePath = PciRoot(0x0)/Pci(0x1,0x1)/Pci(0x0,0x0)  
"
```

hm ...

Beitrag von „anonymous_writer“ vom 21. November 2018, 17:49

Das ist genau das wonach ich gesucht habe.

Versuche gerade was zu finden wo man ganz leicht selber Properties einspielen kann.

Ich bin gerade unterwegs, werde dir dazu aber noch was schreiben

Beitrag von „Si Vis Pacem“ vom 21. November 2018, 18:03

Danke!

Und es eilt wirklich nicht.

Beitrag von „Harper Lewis“ vom 21. November 2018, 19:19

Könnte es per SSDT eventuell so funktionieren (PCI0.PEG1.pci14e4,43b1@0) ?

Code

```
1. DefinitionBlock ("", "SSDT", 2, "ARDARA", "ARPT", 0x00000000)
2. {
3.   External (_SB_.PCI0.PEG1, DeviceObj)
4.   External (_SB_.PCI0.PEG1.ARPT, DeviceObj)
5.
6.   Scope (\_SB.PCI0.PEG1)
7.   {
8.     Device (ARPT)
9.     {
10.    Name (_ADR, Zero) // _ADR: Address
11.    Method(_DSM, 4, NotSerialized)
12.    {
13.      If(LEqual(Arg2, Zero)) { Return(Buffer(){ 0x03 }) }
14.      Return(Package(){
15.        "AAPL,slot-name", Buffer(){ "NGFF M.2 Key A/E" },
16.        "name", Buffer(){ "AirPort Extreme" },
17.        "model", Buffer(){ "Broadcom BCM94352Z - 802.11a/b/g/n/ac Wireless Network Adapter"
18.          },
19.        "device_type", Buffer(){ "AirPort" },
20.        "compatible", Buffer(){ "pci14e4,43a0" },
21.        "built-in", Buffer(){ 0x00 },
22.        "PinConfigurations", Buffer() { 0x00 },
23.      })
24.    }
```

25. }

26. }

Alles anzeigen

Nachtrag: DevicePath = PciRoot(0x0)/Pci(0x1,0x1)/Pci(0x0,0x0) für
PCI0.PEG1.pci14e4,43b1@0, oder?

Beitrag von „Si Vis Pacem“ vom 22. November 2018, 08:42

Müsste ich da nicht zuerst die pcie14r4,43b1 in der SSDT rausnehmen, wie z.B. hier?

Code

1. Scope (B0D4)
2. {
3. Name (_STA, Zero) // _STA: Status
4. }

Dann habe ich wieder das Problem, dass ich pcie14e4,43b1 nicht verwenden kann, da das Komma als Argument interpretiert wird und es einen Fehler gibt.

Update:

Habe es jetzt über den PropertyInjector von [Brumbaer](#) gemacht.

Beitrag von „Harper Lewis“ vom 22. November 2018, 09:20

Hast du es mal ausprobiert mit der SSDT? Müsste gehen, da die selbe Adresse (@0). PropertyInjector ist natürlich auch eine gute Idee. Oder eben direkt in Clover über Devices > Properties.

Nachtrag: Das sollte auch funktionieren:



Beitrag von „Si Vis Pacem“ vom 22. November 2018, 09:33

SSDT geht nicht, da das Komma in pci14e4,43b1 als Argument interpretiert wird und dann ein Fehler geschmissen wird.

Mit PropertyInjector funktioniert das sehr gut.

Beitrag von „Harper Lewis“ vom 22. November 2018, 09:33

Ich meine die SSDT, die ich da oben gepostet habe.

Beitrag von „Si Vis Pacem“ vom 22. November 2018, 09:35

So wie ich die SSDT verstehe, erschafft sie ein Device unter PEG1 ohne vorher das existierende pci14e4,43b1 zu entfernen.

Oder liege ich da jetzt falsch?

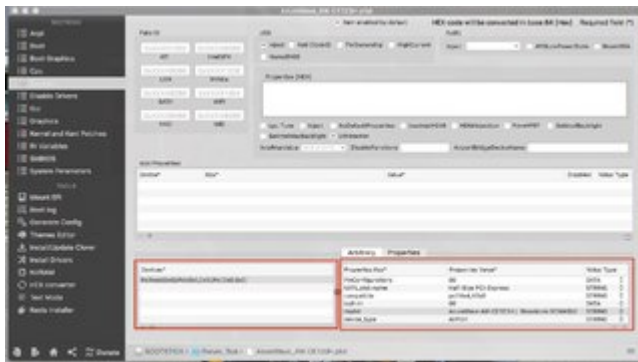
Beitrag von „anonymous_writer“ vom 22. November 2018, 09:35

Genau 3 Methoden gibt es das zu realisieren. 2 hast bereits durch bevor ich Antworten konnte.



Eigentlich wollte ich die Methode 3 Testen da ich da eine Idee habe wie man das Automatisieren könnte. Vielleicht hast auch noch Lust das auch zu testen.

Dazu den ganzen Ast Devices mit Rechtsklick aus der angehängten Vorlage in deine Clover config.plist kopieren.



Beitrag von „Si Vis Pacem“ vom 22. November 2018, 09:37

Probiere ich gleich mal aus.

Beitrag von „anonymous_writer“ vom 22. November 2018, 09:38

Na bist man das Postet gibt es schon wieder 5 Antworten. Cool 😊

Beitrag von „Harper Lewis“ vom 22. November 2018, 09:44

[Si Vis Pacem](#) : Stimmt, das dürfte so nicht funktionieren mit der SSDT. Ich habe aber gerade auch keine Idee, wie man das Device ohne richtigen Device-Name entfernen kann. Hängt das Viech direkt unter PCI0, geht das über die Adresse. Nachtrag: So wie in Beitrag #21 würde ich es auch machen (s. #17). Wenn die PCI-Adresse stimmt, klappt das auf jeden Fall.

Beitrag von „Noir0SX“ vom 22. November 2018, 09:50

[Si Vis Pacem](#) verwendest Du den AirportBrcmFixup Kext ?

Beitrag von „Si Vis Pacem“ vom 22. November 2018, 09:55

[Noir0SX](#)

Jup. Verwende ich.

[anonymous_writer](#)

Hat funktioniert!

Beitrag von „Noir0SX“ vom 22. November 2018, 10:00

Dann Versuche mal die neue Version und dem Bootargument

brcmfx-driver=[0|1|2], 0 - AirPort.BrcmNIC-MFG, 1 - AirPort.Brcm4360, 2 - AirPort.BrcmNIC

Beitrag von „anonymous_writer“ vom 22. November 2018, 10:15

Jetzt sind es 4 Möglichkeiten 🤔👍👎👏

[Harper Lewis](#) , eventuell auch eine einfache Lösung für die DW-1820A.

Wobei interessant ist das oder AirPort.BrcmNIC-MFG, 1 - AirPort.Brcm4360, 2 - AirPort.BrcmNIC.

Bei mir lädt immer AirPort.BrcmNIC-MFG und AirPort.BrcmNIC oder AirPort.Brcm4360.

Beitrag von „Harper Lewis“ vom 22. November 2018, 10:31

Stimmt, vielleicht versuche ich es doch nochmal mit der Karte. Wenn Ein- und Ausbau nur nicht so nervig wären.

Mich würde allerdings trotzdem interessieren, ob das mit der von mir geposteten SSDT funktioniert. Magst du das vielleicht doch mal kurz ausprobieren, [Si Vis Pacem](#) ?

Beitrag von „Noir0SX“ vom 22. November 2018, 10:35

[anonymous_writer](#) vll trifft da noch mehr für Dich zu. Vollständig lautet der Changelog v1.1.7

- Use separate variables and methods wlc_set_countrycode_rev & siPmuFvcoPllreg for

- every driver
- Introduce a new boot-arg `brcmf-driver=[0|1|2]`, 0 - AirPort.BrcmNIC-MFG, 1 - AirPort.Brcm4360, 2 - AirPort.BrcmNIC
 - Remove redundancy boot-args `wl_msg_level` and `wl_msg_level2` (they are supported by Apple)
 - `pci14e4,43b1` & `pci14e4,43b2` matched also to `AirPort_BrcmNIC`
-

Beitrag von „Si Vis Pacem“ vom 22. November 2018, 10:43

[NoirOSX](#)

Ich glaube, ich muss die Device Properties von [anonymous_writer](#) rausnehmen. WLAN nur mit `brcmf-driver=1`. Bei 0 oder 2 "keine hardware installiert".

Bezüglich des ChangeLogs:

Theoretisch müsste ja AirportBrcmFixup jetzt alles übernehmen ohne dass ich FakeID oder einen sonstigen Injector nehme, oder verstehe ich das falsch?

Werde es mal so versuchen.

Beitrag von „anonymous_writer“ vom 22. November 2018, 11:07

Habe auch die Parameter `brcmf-driver=[0|1|2]` bei mir getestet mit dem gleiche Ergebnis nur `brcmf-driver=1` funktioniert.

Der wichtige Parameter "compatible" = "pci14e4,43a0" wird dadurch leider nicht gesetzt.

Beitrag von „Si Vis Pacem“ vom 22. November 2018, 11:16

Bin jetzt mal in die Info.plist rein und da liegt wohl das Problem.

b1 ist weiterhin unter 4360 und nicht unter NIC.

Ausgeschnitten und unter NIC und es funktioniert wie zuvor.

Ich nehme an, dass ein Teil von AirportFixup, der den brcmfxf-driver Eintrag übernimmt, funktioniert, die Zuordnung zu NIC aber nicht. Daher dann kein Treiber geladen wird.