

Erledigt

Thermal subsystem, was ist das und wie kann man das nutzen?

Beitrag von „anonymous_writer“ vom 31. August 2018, 08:22

Hallo liebe Forenteilnehmer,

Durch den PropertyInjector.kext von [@Brumbaer](#) und [@NoirOSX](#) tun sich ganz unerwartete Dinge auf und man findet Devices welche einem vorher noch nicht mal aufgefallen sind.

So ist es auch mit dem "Thermal subsystem". [Hier ein link zu meinem.](#)

Was macht ein "Thermal subsystem"?

Es ist ein allgemeiner Mechanismus um das Wärmemanagement in verschiedenen Zonen zu steuern. Normalerweise besteht es aus einer oder mehreren thermischen Zonen und Kühlvorrichtungen.

Für Windows und auch Linux kann man dazu Treiber finden über welche man dieses System ansprechen kann.

[Hier ein Link zum Linux Treiber.](#)

Nun zu OSX:

Unter OSX ist es mir möglich das Device unter Systeminformationen PCI anzeigen zu lassen. Nur leider wird dazu kein Treiber geladen.

Drei Möglichkeiten fallen mir dazu ein warum das so ist:

> "Thermal subsystem" wird vom Kernel unterstützt. Wäre der optimale Weg

> "Thermal subsystem" gibt es bei Apple nicht und wird daher auch nicht unterstützt.

> "Thermal subsystem" auf meinem Laptop ist abweichend zum Apple "Thermal subsystem" und muss bei einem Hacki entsprechend gefaket werden.

Genau an dieser Stelle komme ich nicht weiter und schreibe daher diese Zeilen. Eventuell hat jemand mehr Ahnung und kann zu diesem Thema weiterhelfen.

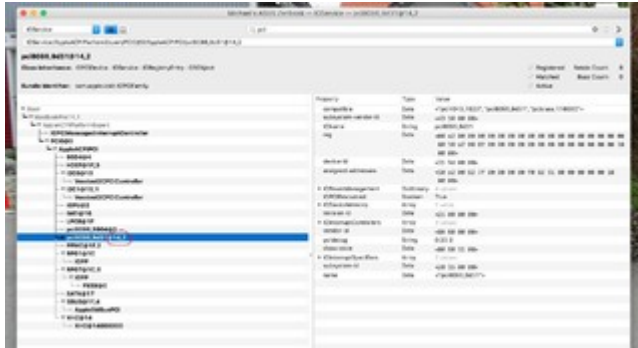
Für alle die ein "Thermal subsystem" besitzen hier eine kleine Anleitung wie ich das Device unter PCI sichtbar machen kann.

Auf meinem Laptop fehlt dieses Device bereits in der DSDT. Daher habe ich eine eigene Device THRC erzeugt. THRC ist frei erfunden und soll heißen "Thermal Controller".

Über "Name(_ADR, 0x00140002)" wird das Device angesprochen.

Wie kommt man zu dem richtigen Wert von _ADR?

Über den [IORegistryExplorer](#). Diesen starten und dann nach PCI suchen. Bei mir findet man das "Thermal subsystem" unter "pci8086,9d31@14,2".



Wichtig ist die Info 14,2. Der Wert ergibt dann den Eintrag "Name(_ADR, 0x00140002)" mit welcher man dieses Device ansprechen kann.



PS: die THRC.dsl muß am Ende mit dem [MaciASL](#) als THRC.aml abgespeichert werden und dann nach "\EFI\CLOVER\ACPI\patched" kopiert werden.

Beitrag von „FARV“ vom 31. August 2018, 10:18

Moin,

das Ganze klingt sehr interessant!

Damit könnte man die ganzen FAN-Header der Mainboards steuern und ggf. ähnlich wie mit dem smcfancontrol Einfluss auf die Lüftersteuerung ausüben.

Wenn man das Ganze zudem noch richtig aufzieht, dann könnte man sogar eigene Lüfterkurven anlegen und nicht nur zwischen "auto" und "manuell" mit deinem Drehzahlregler wählen.

Für Windows-PC's gibt es so ein Tool von nahezu jedem Mainboard-Hersteller und auch "freie" Tools sind haufenweise vorhanden.

Für Mac/Hacks vermisste ich so ein Tool (in dem Umfang) allerdings.



Das wäre doch mal ein Projekt für die Programmierer hier im Forum?!

Oder interpretiere ich das alles hier gerade falsch?

Gruß

Tim

Beitrag von „anonymous_writer“ vom 31. August 2018, 10:49

Hallo [@FARV](#),

das ist schon richtig. Eine entscheidende Frage dazu ist ob Apple ein solches Sub System bei den Real Apple Geräten überhaupt unterstützt. Dazu ist leider nichts zu finden.

Beitrag von „FARV“ vom 31. August 2018, 11:28

Keine Ahnung, ob es ein regelrechtes Subsystem in echten Macs gibt, aber Apple verbaut haufenweise Temperaturfühler und co. in seinen Geräten. Diese werden ja auch von smcfancontrol oder HWMonitor ausgelesen und angezeigt.

Zudem bessert Apple immer mal wieder die Lüftersteuerung von den Rechnern per Firmware-Update nach. Jetzt bin ich kein Spezi in diesem Bereich, aber für mich klingt das schon so, als wenn da so etwas wie ein Thermal Subsystem am Gange ist. Doch dazu sollten vielleicht einmal die Spezis was sagen!

Gruß
Tim

Beitrag von „apfelnico“ vom 31. August 2018, 12:51

Läuft. Schreibe gleich etwas dazu. Danke für den entscheidenden Ansatz.

Edit:

Wie auf Bild "001" zu sehen ist, gibt es ein "Thermal Subsystem". "pci8086,a2b1" kennt macOS nicht, aber interessanterweise gibt es ebenfalls im Bild "001" gleich darunter einen alten Bekannten, den "PMCR", bzw "Power Management Controller". Der hat nur eine kleine Abweichung, "pci8086,a2a1", also höchstwahrscheinlich eine ähnliche Funktion. Dieser ist bei mir erfolgreich ins System integriert.

Also einfach mal probiert, der entscheidende Eintrag ist bei mir "compatible" 😊

Code

```
1. Scope (\_SB.PCI0)
2. {
3.   Device (THSS)
4.   {
5.     Name (_ADR, 0x00140002) // _ADR: Address
6.     Name (THSS, One)
7.     Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
8.     {
9.       Store (Package (0x0C)
10.      {
11.        "AAPL,slot-name",
12.        Buffer (0x09)
13.        {
14.          "Built In"
15.        },
16.
17.
18.        "built-in",
19.        Buffer (One)
20.        {
21.          0x00
22.        },
23.
24.
25.        "name",
26.        Buffer (0x24)
27.        {
28.          "Intel X299 Series Thermal Subsystem"
29.        },
30.
31.
32.        "model",
33.        Buffer (0x24)
34.        {
35.          "Intel X299 Series Thermal Subsystem"
36.        },
37.
38.
```

```
39. "device_type",
40. Buffer (0x13)
41. {
42. "Thermal-Controller"
43. },
44.
45.
46. "compatible",
47. Buffer (0x0D)
48. {
49. "pci8086,a2a1"
50. }
51. }, Local0)
52. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
53. Return (Local0)
54. }
55. }
56. }
```

Alles anzeigen

Viel Spaß

Beitrag von „anonymous_writer“ vom 31. August 2018, 12:56

Voll cool wenn das geht. Teste das dann später. Vielen Dank für deine Info 👍

Beitrag von „apfelnico“ vom 31. August 2018, 13:15

Bei dir wäre der compatible Eintrag "pci8086,9d21".
Darauf matcht "com.apple.driver.AppleIntelPCHPMC" ebenso.

Beitrag von „anonymous_writer“ vom 31. August 2018, 13:26

Das Device THSS fehlt in meiner DSDT.aml. Kannst du mir deine DSDT.aml zur Verfügung stellen. Möchte mal versuche ob ich das einfach kopieren kann oder anhand deiner Einträge bei mir in der DSDT.aml suchen ob das irgendwie anders heißt.

Beitrag von „apfelnico“ vom 31. August 2018, 13:32

Sorry, das habe ich ebenfalls nicht. Ist in meiner SSDT jetzt neu drin, worin ich auch sonstige Geräte näher spezifiziere. Fand nur den Namen "THSS" passender. 😊
Wenn in eine eigene SSDT, dann darauf achten, dass im Header auch der Verweis auf die externe Methode "DTGP" gesetzt wird.

Edit:

Du kannst deine SSDT nutzen, nur noch in der _DSM-Methode den "compatible"-Eintrag "pci8086,9d21" einfügen ...

Beitrag von „anonymous_writer“ vom 31. August 2018, 13:37

Verstanden, das werde ich Testen. 👍

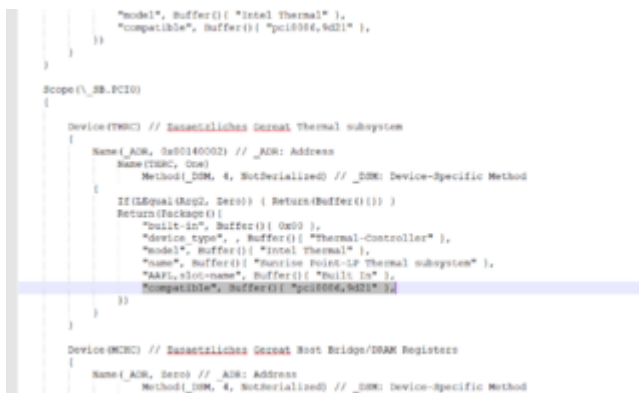
Beitrag von „anonymous_writer“ vom 31. August 2018, 13:59

Hallo @apfelnico,

Ich glaub es fast nicht 😊 , aber das war der Trick. "pci8086,9d21" war der entscheidende Hinweis und das gleich bei beiden Devices. 😄



Ich Danke für deine Untersützung.



Beitrag von „apfelnico“ vom 31. August 2018, 14:06

Welches zweite Device? HECI? Das solltest du nur einfach umbenennen in "EMEI". Ach so, du meinst den PMC.

Beitrag von „anonymous_writer“ vom 31. August 2018, 14:13

Das hier im Bild. Dazu gibt es auch ein DSDT Device, hat aber bis jetzt nicht funktioniert.


```

287
288
289 // Range of vR3-R1020 // Range R3-R1020 vR3-R100 vR30 Gen Core Processor Thermal Subsystem
290
291 {
292     Method(_DSM, 4, NotSerialized)
293     {
294         if (IsApple (Arg0, 0)) { ReturnBuffer(1); }
295         ReturnBuffer(0);
296         "MAIL_SLOT_NAME", Buffer(1) "Mail Slot 1";
297         "Mail Slot", Buffer(1) "Mail Slot 1";
298         "Service Type", Buffer(1) "Thermal-Controller";
299         "Model", Buffer(1) "Intel Thermal";
300         "Compatible", Buffer(1) "pci1000a,8086" [];
301     }
302 }
303

```

Wird mit dem gleiche Trick geladen. 😁

Beitrag von „DSM2“ vom 31. August 2018, 15:44

Hat es abseits von Kosmetik auch einen nutzen, hat das jemand von euch bereits testen können ?

Beitrag von „anonymous_writer“ vom 31. August 2018, 16:08

Denke schon das es einen nutzen hat da jetzt der Power Management Controller Kext AppleIntelPCHPMC für diese beiden Device geladen wird.

Welchen kann ich dir noch nicht sagen da das ganze erst seit heute für mich existiert und funktioniert.



Beitrag von „Harper Lewis“ vom 31. August 2018, 18:38

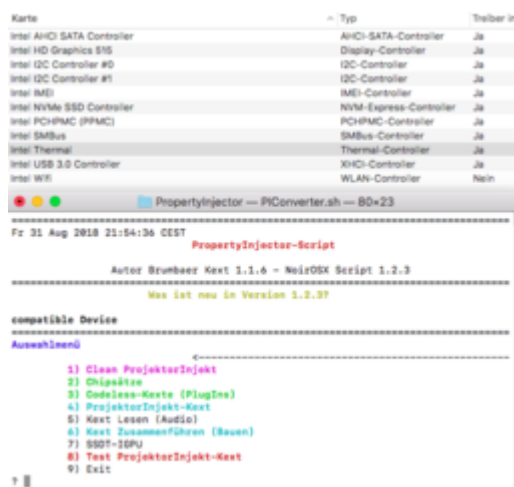
Das funktioniert auf meinem Dell Vostro 5370 auch (pci8086,9d31@14,2). Ein Device B0D4@4 sehe ich ebenfalls, es ist [dieses](#) hier.

Beitrag von „Noir0SX“ vom 31. August 2018, 22:00

Danke [@anonymous writer](#) das Du es hier nochmal in einen extra Beitrag geschrieben hast und [@apfelnico](#) das dann umgesetzt hat.

Mal sehen was es bewirkt.

Bei mir wird er nun auch geladen



```
Karte          Typ          Treiber in
Intel AHCI SATA Controller  AHCI-SATA-Controller  Ja
Intel HD Graphics 515       Display-Controller     Ja
Intel I2C Controller #0     I2C-Controller         Ja
Intel I2C Controller #1     I2C-Controller         Ja
Intel IMEI                  IMEI-Controller        Ja
Intel NVMe SSD Controller   NVMe-Express-Controller Ja
Intel PCHPMC (PPMC)         PCHPMC-Controller      Ja
Intel SMBus                 SMBus-Controller       Ja
Intel Thermal               Thermal-Controller     Ja
Intel USB 3.0 Controller    XHCI-Controller        Ja
Intel WLAN                  WLAN-Controller        Nein

PropertyInjector - PCConverter.sh - 80x23

Fr 31 Aug 2018 21:54:36 CEST

PropertyInjector-Script

Autor Brumbeer Kart 1.1.6 - Noir0SX Script 1.2.3

Was ist neu in Version 1.2.3?

kompatible Device
Auswählen()

1) Clean ProjektorInject
2) Chipset
3) Codeless-Karte (PlugIns)
4) ProjektorInject-Kart
5) Kart Lesen (Audio)
6) Kart Zusammenführen (Bauen)
7) SSDT-IGPU
8) Test ProjektorInject-Kart
9) Exit
```

Beitrag von „CMMChris“ vom 31. August 2018, 23:13

Interessant, auf meinem Z370er Aorus Ultra Gaming ist dergleichen nicht zu finden. IOJones listet hier nur zwei unbekannte Geräte "pci8086,3ec2@0" und "pci8086,1911@8". Die haben

aber was mit USB zu tun - vermutlich meine nicht injecteten 3.1er Ports.

Beitrag von „the_viking90“ vom 1. September 2018, 09:19

Moment...ich komm grade nicht mehr mit.
Wofür benutzt ihr diesen PropertyInjector.kext?

ich steig da grade nicht hinter.

Beitrag von „apfelnico“ vom 1. September 2018, 09:36

[@the_viking90](#)

du kannst Eigenschaften für Geräte in DSDT, SSDT und/oder in einer Kext definieren.

Beitrag von „Noir0SX“ vom 1. September 2018, 09:44

... und in der config.plist

Beitrag von „the_viking90“ vom 1. September 2018, 10:45

Und das kann man parallel dann zu DSDT etc laufen lassen?

Beitrag von „revunix“ vom 1. September 2018, 10:53

Ich blick da auch nicht wirklich durch, geht das ganze auch bei alten Mainboards oder nur X299'er ?

Beitrag von „kuckkuck“ vom 4. September 2018, 14:46

[Zitat von the_viking90](#)

Und das kann man parallel dann zu DSDT etc laufen lassen?

Kann man... Manche Dinge können aber nur per DSDT und nicht PropertyInjector injected werden, da die DSDT früher "greift".

[Zitat von Un!x](#)

Ich blick da auch nicht wirklich durch, geht das ganze auch bei alten Mainboards oder nur X299'er ?

Die PropertyInjector.kext ist für jegliche Systeme geeignet. Zum eigentlichen Thema dieses Threads: Ob das bei deinem Mainboard geht kommt drauf an, ob dein Mainboard ein Thermal subsystem verbaut hat. Bei meinen älteren Chipsets ist das nirgendwo der Fall...

Beitrag von „the_viking90“ vom 8. September 2018, 01:32

Wie bekommt man das raus?

Beitrag von „iPhoneTruth“ vom 26. Januar 2019, 16:00

Super, das hat bei mir auch geklappt. Und zwar einfach dadurch, daß ich in der Info.plist im

PropertyInjector.kext den Eintrag compatible hinzugefügt habe.

Seither läuft die Kiste merkbar ruhiger, die Lüfter drehen bei weitem nicht so hoch wie früher!

Mei, seid's Ihr guad !!!



Info.plist
Jetzt kaufen

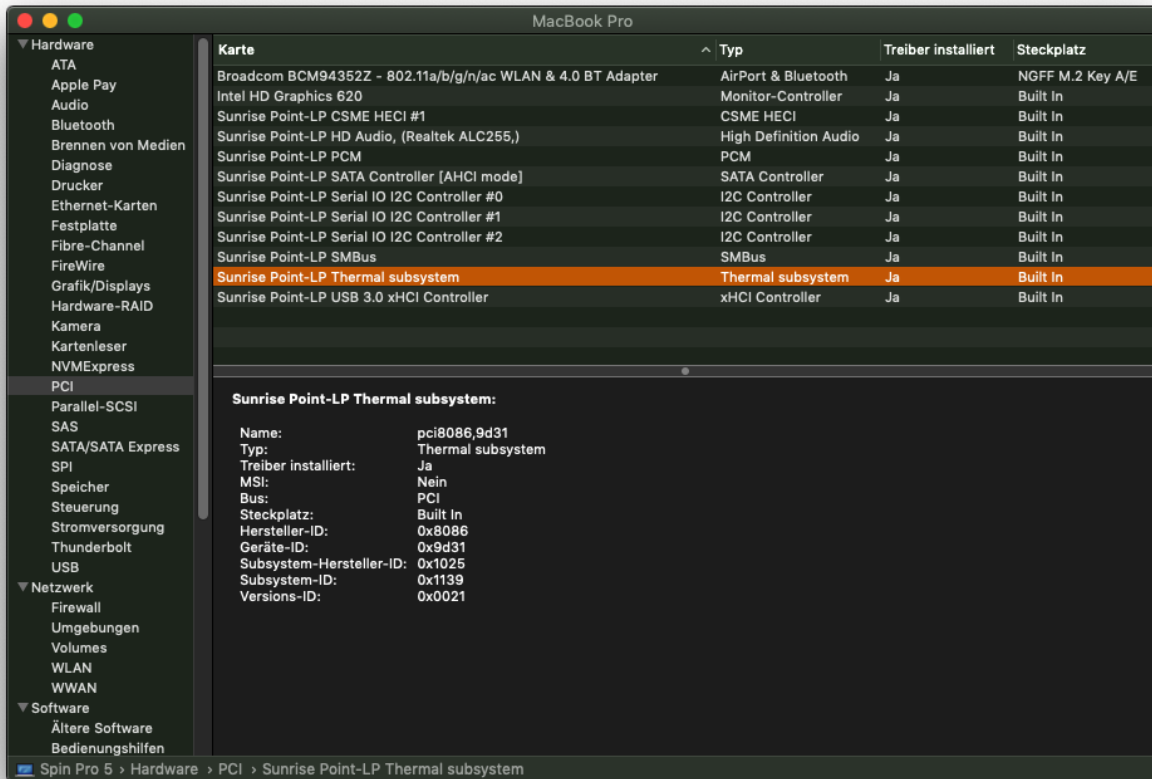
Neues Kind-Element
 Neues Geschwister-Element
 Löschen

Property-List	Type	Wert
CFBundleName	String	PropertyInjector
CFBundlePackageType	String	KEXT
CFBundleShortVersionString	String	1.1
CFBundleSupportedPlatforms	Array	1 geordnete Elemente
CFBundleVersion	String	1
DTCompiler	String	com.apple.compilers.llvm.clang.1_0
DTPlatformBuild	String	9C40b
DTPlatformVersion	String	GM
DTSDKBuild	String	17C76
DTSDKName	String	macosx10.13
DTXcode	String	0920
DTXcodeBuild	String	9C40b
IOKitPersonalities	Dictionary	13 Schlüssel/Wert-Paare
Audio-ALC255	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP CSME HECI #1	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP LPC Controller	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP PCI Express Root Port #7	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP PCM	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP SATA Controller [AHCI mode]	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP SMBus	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP Serial IO I2C Controller #0	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP Serial IO I2C Controller #1	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP Serial IO I2C Controller #2	Dictionary	5 Schlüssel/Wert-Paare
Sunrise Point-LP Thermal subsystem	Dictionary	5 Schlüssel/Wert-Paare
CFBundleIdentifier	String	de.Brumbaer.PropertyInjector
IOClass	String	PropertyInjector
IOPCIPrimaryMatch	String	0x9d318086
IOPProviderClass	String	IOPCIDevice
Properties	Dictionary	4 Schlüssel/Wert-Paare
AAPL,slot-name	Daten	9 Bytes: 4275696C 7420496E 00
compatible	Daten	ASCII-Text: pci8086,9d21
device_type	Daten	17 Bytes: 54686572 6D616C20 73756273 79737465 6D
model	Daten	34 Bytes: 53756E72 69736520 5...616C 20737562 73797374 656D
Sunrise Point-LP USB 3.0 xHCI Controller	Dictionary	5 Schlüssel/Wert-Paare
Xeon E3-1200	Dictionary	5 Schlüssel/Wert-Paare
NSHumanReadableCopyright	String	Copyright © 2017 Brumbaer. All rights reserved.
OSBundleLibraries	Dictionary	5 Schlüssel/Wert-Paare

Strukturansicht synchronisieren ↑
☒ Text automatisch synch
Text synchronisieren ↓

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>BuildMachineOSBuild</key>
  <string>17D29a</string>
  <key>CFBundleDevelopmentRegion</key>
  <string>en</string>
  
```

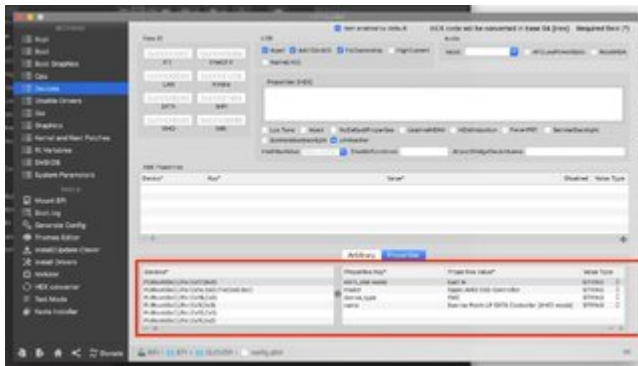


Beitrag von „an3k“ vom 26. Januar 2019, 22:34

Ich muss da jetzt mal einhaken. Wie bekommt ihr die Liste in PCI hin? Bei mir ist die komplett leer obwohl zumindest Grafikkarte und Audio, als auch AHCI und NVMe installiert sind und funktionieren.

Beitrag von „anonymous_writer“ vom 26. Januar 2019, 22:59

Es gibt mehrere Möglichkeiten dazu. Eine ist über Clover > Devices > Properties.



Kannst dir hier mal ansehen:

<https://bitbucket.org/anonymou...r/EFI/CLOVER/config.plist>

Die richtigen Adressen bekommt man über diese Tool:

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

Beitrag von „CMMChris“ vom 26. Januar 2019, 23:21

Aber Achtung! Das Angeben des Slot Name kann auch Probleme machen. Bei mir killt es zum Beispiel Audio wenn ich die Soundkarte mit einem Slot Name versehe. Ich persönlich habe mich deshalb entschieden die Liste leer zu lassen. Ist eh nur Kosmetik.

Beitrag von „iPhoneTruth“ vom 27. Januar 2019, 18:42

[CMMChris](#) Umkehrt ist es bei mir. Dadurch, daß ich die PCI-Geräte sichtbar gemacht habe, ist mir aufgefallen, daß das Thermal subsystem keinen Treiber findet, wodurch mir klar wurde, weshalb der Lüfter so oft auch anging. Nun konnte ich das beheben. Somit kann es also auch seinen Sinn haben und manchmal nicht nur "reine Kosmetik" sein.

[anonymous_writer](#) Ich habe das über die Properties in der config.plist mit gfxutil auch probiert,

die PCI-Geräte zeigt er damit auch an, allerdings hat der Eintrag "compatible" über diese Methode keinen Effekt, so daß dann angegeben wird, daß der Treiber für das Thermal subsystem nicht geladen wird. Über den PropertyInjector klappt das dagegen.

Beitrag von „CMMChris“ vom 27. Januar 2019, 18:46

[iPhoneTruth](#) Dass es hilfreich sein kann die Geräte mal sichtbar zu machen bestreite ich ja nicht. Aber die Liste an sich ist reine Kosmetik. Das reine Auftauchen in der Liste hat ja keine funktionalen Auswirkungen, außer es bringt was durcheinander, was wie gesagt passieren kann.

Beitrag von „anonymous_writer“ vom 27. Januar 2019, 18:48

Hallo [iPhoneTruth](#) ,

ja das ist richtig. Dem Thermal subsystem fehlt die Methode im Bios. Daher muss dafür zusätzlich eine Methode in einer SSDT erstellt werden. Erst dann funktioniert das mit dem "compatible" .

Beitrag von „iPhoneTruth“ vom 28. Januar 2019, 09:36

Bei mir funktioniert es ohne zusätzliche SSDT einfach nur mit dem "compatible" im PropertyInjector.kext

Beitrag von „anonymous_writer“ vom 28. Januar 2019, 09:42

Interessant, aber dann sollte es eigentlich auch über die Properties in der config.plist funktionieren. Die machen genau das gleiche.

Hauptsache es funktioniert. 👍

Beitrag von „iPhoneTruth“ vom 28. Januar 2019, 09:50

Ja, ich hatte auch gedacht, daß es über die config.plist eigentlich auch gehen müßte. Aber nach zwei Versuchen bin ich zum Ergebnis gekommen, daß ... sagen wir es so ... ich es über Properties in der config.plist nicht hinbekomme 🤔 !

Vielleicht oder wahrscheinlich liegt das an dem zusätzlichen Code, der in der Kext von [Brumbaer](#) vorhanden ist.

Auf jeden Fall macht es das Leben für mich einfacher, da ich den Umgang mit dem Erstellen, Anpassen und Ändern von SSDT's noch nicht so richtig beherrsche. Kommt vielleicht noch 😊 !

Beitrag von „CMMChris“ vom 20. April 2019, 00:57

Weil ich gerade zufällig nochmal auf dieses Thema gestoßen bin: Mittlerweile wird zumindest der a2a1 PMC ohne jegliche SSDT oder sonstiges erkannt und genutzt.