

Sonnet M.2 4x4 PCIe Card via SSDT als internes Gerät Konfigurieren - Sage 10/G

Beitrag von „pgr69“ vom 1. November 2020, 17:05

Hallo [apfelnico](#)

Wie kann ich eine Sonnet Fusion SSD M.2 4x4 die als "Extern" angezeigt wird Intern deklarieren?

Unter PC01.BR1A.SL01.pci-bridge@8 finde ich dann die besagten Laufwerke. VendorID ist wieder 144d für Samsung und DeviceID a808 für die 970Pro.



Kann ich nun das von DSM2 gepostete Beispiel nehmen und `_SB.PC00.RP21.PXSX` ersetzen durch `_SB.PC00.BR1A.SL01` ersetzen? Wobei die Samsung IDs ja erst unter `PC01@0/BR1A@0/SL01@0/pci-bridge@8/pci-bridge@0/pci-bridge@8/pci144d,a808@0` auftauchen. Alles davor sind die Multiplexer für die PCI-Lanes (Vendor 10b5 Device 8747 und 8732)

Wie ist das mit der UUID...nehme ich da die selbe wie im Beispiel oder ist das bei mir eine andere ?

Ig und Danke

Beitrag von „apfelnico“ vom 1. November 2020, 17:45

[Zitat von pgr69](#)

Also würde das Beispiel von DSM2 dazu führen dass eine in dem M.2 Anschluss angeschlossene PCIe-NVMe entsprechend angezeigt wird. Da die SM951 aber eine SATA-SSD im M.2 Format ist erscheint diese im Systembericht unter SATA und nicht unter NVME.

Zum einen ist es eben ein BEISPIEL, zum anderen hast du es immer noch nicht verstanden. Im Systembericht unter PCI würdest du einen Eintrag sehen, so wie du das Gerät unter "name" eben benennst und als Slot eben "M.2". Völlig Wurscht, ob das nun NVMe oder SATA ist. Entsprechend siehst du bei NVMe natürlich nix. Würdest du in der SSDT bei "name" von mir aus "tolles Gerät" schreiben und für den Slot "der kleine" schreiben, dann würde eben das angezeigt werden.

[Zitat von pgr69](#)

MEIN PROBLEM liegt ja nicht ...

Das hat hier jeder verstanden. Du anscheinend nicht, dass es nur um ein Beispiel ging, dass lediglich die Syntax erklärt.

[Zitat von pgr69](#)

Unter PC01.BR1A.SL01.pci-bridge@8 finde ich dann die besagten Laufwerke.

Bingo. In deiner SSDT gibst du nun also diese Adresse ein, anstelle von dem derzeit noch nicht in der ACPI vorhandenem Device "pci-bridge@8" denkst du dir ein neues (vier Zeichen) Device aus. Sofern deine Sonnet eben auf "pci-bridge@8" liegt. Liegt sie davor (SL01), dann beschreibst du eben diesen vorhandenen Scope. Anbei mal ein Versuch aus der Ferne, ohne weitere Hintergründe:

Code

1. DefinitionBlock ("", "SSDT", 2, "NICO", "SOFU", 0x00900000)
2. {
3. External (_SB_.PC01.BR1A.SL01, DeviceObj)
- 4.
5. Scope (_SB.PC01.BR1A.SL01)
6. {
7. If (_OSI ("Darwin"))

```

8. {
9. Method (DTGP, 5, NotSerialized)
10. {
11. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
12. {
13. If ((Arg1 == One))
14. {
15. If ((Arg2 == Zero))
16. {
17. Arg4 = Buffer (One)
18. {
19. 0x03 // .
20. }
21. Return (One)
22. }
23.
24. If ((Arg2 == One))
25. {
26. Return (One)
27. }
28. }
29. }
30.
31. Arg4 = Buffer (One)
32. {
33. 0x00 // .
34. }
35. Return (Zero)
36. }
37.
38. Device (SOFU)
39. {
40. Name (_ADR, 0x00080000) // _ADR: Address
41. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
42. {
43. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
44. {
45. Local0 = Package (0x08)
46. {
47. "AAPL,slot-name",
48. Buffer (0x07)

```

```
49. {
50. "Slot-2"
51. },
52.
53. "built-in",
54. Buffer (One)
55. {
56. 0x00 // .
57. },
58.
59. "model",
60. Buffer (0x12)
61. {
62. "Sonnet Fusion SSD"
63. },
64.
65. "device_type",
66. Buffer (0x17)
67. {
68. "NVM Express-Controller"
69. }
70. }
71. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
72. Return (Local0)
73. }
74.
75. Return (Zero)
76. }
77. }
78. }
79. }
80. }
```

Alles anzeigen

Du siehst, dass hier "hinter" SL01 nun eine neues Device "SOFU" (Sonnet Fusion) eingefügt wurde, mit einer spezifischen Adresse "_ADR, 0x00080000" (pci-bridge@8).

[Zitat von pgr69](#)

Und nochmal zur UUID...nehme ich da die selbe wie im Beispiel oder ist das bei mir

eine andere ?

Nein, wieso? Ich schrieb doch, es referenziert auf die Methode "DTGP". Du solltest dir schon die Mühe machen, diese Methode auch mal anzuschauen, dann fällt dir sicher etwas auf. In meinem Beispiel habe ich diese Methode integriert.

[Zitat von pgr69](#)

Kann ich nun das von DSM2 gepostete Beispiel nehmen und `_SB.PC00.RP21.PXSX` ersetzen durch `_SB.PC00.BR1A.SL01` ersetzen? Wobei die Samsung IDs ja erst unter `PC01@0/BR1A@0/SL01@0/pci-bridge@8/pci-bridge@0/pci-bridge@8/pci144d,a808@0` auftauchen. Alles davor sind die Multiplexer für die PCI-Lanes (Vendor 10b5 Device 8747 und 8732)

Es geht NICHT um die angeschlossenen NVMe an den vier Steckplätzen. Es geht um den Sonnet-Controller davor. Der sollte, wie im Beispiel beschrieben, als "internes Gerät" definiert werden.

Beitrag von „pgr69“ vom 1. November 2020, 18:56

Hi

[apfelnico](#):

Doch ich habe das schon so weit verstanden auch wenn es nicht so aussah. Wir haben da wohl aneinander vorbei geredet.

Das mit UUID kam mir eben etwas seltsam vor deshalb noch mal die Nachfrage.

Bin jetzt dran und berichte.

Ig und DANKE

SO [apfelnico](#)

Das hat leider nicht funktioniertalso es hat schon funktioniert aus dem Device nu nbdas Device SOFU zu machen, aber meine Laufwerke sind immer noch extern.

Die in Deinem Beispiel verwendete Adresse 0x00080000....ist die erfunden oder woher kommt die?

Wenn ich versuche tiefer einzusteigen als noch pci-bridge anhänge dann lässt es sich nicht kompilieren.

Ig

Beitrag von „apfelnico“ vom 1. November 2020, 20:31

[Zitat von pgr69](#)

Die in Deinem Beispiel verwendete Adresse 0x00080000....ist die erfunden oder woher kommt die?

Nein. Wieso sollte ich die "erfinden"?

[Zitat von apfelnico](#)

spezifischen Adresse "_ADR, 0x00080000" (pci-bridge@**8**).

Zeig mal deine IORegistry mit geladener SSDT. Hast du meinen Quellcode 1:1 übernommen? Was ist im Systembericht/Systeminformationen zu sehen?

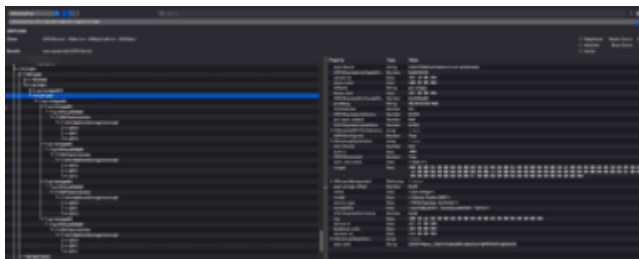
Beitrag von „pgr69“ vom 1. November 2020, 20:42

[apfelnico](#)

Ich meinte ob diese Adresse beliebig ist und wenn nicht woraus resultiert die. Dazu passen würden zahlen die jeweils hinter den Namen mit dem @-Zeichen folgen und das nach hinten auf 8 Stellen mit nullen aufgefüllt: IODeviceTree:/PC01@0/BR1A@0/SL01@0/SOFU@8

Ja konnte den Code einwandfrei übernehmen. Habs dann noch in die config.plist eingetragen und dann hat er es auch geladen.

Hier die Screenshots



| | | | |
|--------------|------------------------|----|-----------------|
| pci144d,a808 | NVM Express Controller | Ja | Slot- 2@166,0,0 |
| pci144d,a808 | NVM Express Controller | Ja | Slot- 2@165,0,0 |
| pci144d,a808 | NVM Express Controller | Ja | Slot- 2@168,0,0 |
| pci144d,a808 | NVM Express Controller | Ja | Slot- 2@167,0,0 |

| | | | |
|--------------------------|------------------------|----|-----------------|
| Intel X550T Ethernet 100 | Ethernet-Controller | Ja | Slot In |
| Intel X550T Ethernet 100 | Ethernet-Controller | Ja | Slot In |
| pci144d,a808 | NVM Express Controller | Ja | Slot- 2@166,0,0 |
| pci144d,a808 | NVM Express Controller | Ja | Slot- 2@165,0,0 |
| pci144d,a808 | NVM Express Controller | Ja | Slot- 2@168,0,0 |
| pci144d,a808 | NVM Express Controller | Ja | Slot- 2@167,0,0 |
| pci1002,aa18 | Audgerät | Ja | Slot- 6@197,0,1 |
| Radeon RX Vega 64 | Monitor-Controller | Ja | Slot- 6@197,0,0 |

| | | | |
|--------------------------|------------------------|--|--|
| pci144d,a808: | | | |
| Typ: | NVM Express Controller | | |
| Treiber installiert: | Ja | | |
| MD: | Ja | | |
| Bus: | PCI | | |
| Steckplatz: | Slot- 2@166,0,0 | | |
| Hersteller-ID: | 0x144d | | |
| Geräte-ID: | 0xa808 | | |
| Subsystem-Hersteller-ID: | 0x144d | | |
| Subsystem-ID: | 0xa801 | | |
| Version-ID: | 0x0000 | | |
| Seriennummer: | S483NXDMA08E2EW | | |
| Link-Breite: | x4 | | |
| Link-Geschwindigkeit: | 8.0 GT/s | | |

Beitrag von „apfelnico“ vom 1. November 2020, 20:54

[pgr69](#)

Nein, es ist "nur" die "@8". Und wahrscheinlich falsch, weil wir damit die "Bridge" beschrieben haben, aber nicht den Controller an sich. Wie ich oben schon schrieb, eventuell das Device davor (SL01), beziehungsweise in einem "Scope" (einem vorhandenen Device weitere Properties hinzufügen). Also noch mal, nun so:

Code

```
1. DefinitionBlock ("", "SSDT", 2, "NICO", "SOFU", 0x00900000)
2. {
3.   External (_SB_.PC01.BR1A.SL01, DeviceObj)
4.
5.   Scope (\_SB.PC01.BR1A.SL01)
6.   {
7.     If (_OSI ("Darwin"))
8.     {
9.       Method (DTGP, 5, NotSerialized)
10.    {
11.      If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
12.      {
13.        If ((Arg1 == One))
14.        {
15.          If ((Arg2 == Zero))
16.          {
17.            Arg4 = Buffer (One)
18.            {
19.              0x03 // .
20.            }
21.            Return (One)
22.          }
23.
24.          If ((Arg2 == One))
25.          {
26.            Return (One)
27.          }
28.        }
29.      }
30.
31.      Arg4 = Buffer (One)
32.      {
33.        0x00 // .
34.      }
```



```

35. Return (Zero)
36. }
37.
38. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
39. {
40. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
41. {
42. Local0 = Package (0x08)
43. {
44. "AAPL,slot-name",
45. Buffer (0x07)
46. {
47. "Slot-2"
48. },
49.
50. "built-in",
51. Buffer (One)
52. {
53. 0x00 // .
54. },
55.
56. "model",
57. Buffer (0x12)
58. {
59. "Sonnet Fusion SSD"
60. },
61.
62. "device_type",
63. Buffer (0x17)
64. {
65. "NVM Express-Controller"
66. }
67. }
68. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
69. Return (Local0)
70. }
71.
72. Return (Zero)
73. }
74. }
75. }

```

76. }

Alles anzeigen

Beitrag von „pgr69“ vom 1. November 2020, 21:09

[apfelnico](#)

Okay also diesmal kein neues Device deklarieren sondern dem vorhandenen SL01 die Properties zuweisen.....hat leider auch nichts gebracht. Die Properties wurden übernommen aber die SSDs sind immer noch "externe".

Kann man das nicht als property direkt auf die 4 einzelnen SSDs bzw. deren Controller schreiben?

Sobald ich versuche hinter SL01 weiter zu gehen gibt es Fehler...alsoSL01.pci-bridge.pci-bridge mag er nicht

Oder muss jetzt erst die hintereinander liegenden PCI-Bridges nacheinander als neue Devices definiert werden damit ich ihnen Properties zuweisen kann ?

Ig und Danke

Beitrag von „apfelnico“ vom 1. November 2020, 21:35

[Zitat von pgr69](#)

Oder muss jetzt erst die hintereinander liegenden PCI-Bridges nacheinander als neue Devices definiert werden damit ich ihnen Properties zuweisen kann ?

Genau. Anders geht es nicht via SSDT. Es muss ja ein Device vorhanden sein, das man beschreiben will. Noch mal den Baum genau angeschaut, dann eben ganz tief rein:

Code

```
1. DefinitionBlock ("", "SSDT", 2, "NICO", "SOFU", 0x00900000)
2. {
3. External (_SB_.PC01.BR1A.SL01, DeviceObj)
4.
5. Scope (\_SB.PC01.BR1A.SL01)
6. {
7. If (_OSI ("Darwin"))
8. {
9. Method (DTGP, 5, NotSerialized)
10. {
11. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
12. {
13. If ((Arg1 == One))
14. {
15. If ((Arg2 == Zero))
16. {
17. Arg4 = Buffer (One)
18. {
19. 0x03 // .
20. }
21. Return (One)
22. }
23.
24. If ((Arg2 == One))
25. {
26. Return (One)
27. }
28. }
29. }
30.
31. Arg4 = Buffer (One)
32. {
33. 0x00 // .
34. }
35. Return (Zero)
36. }
37.
38. Device (A001)
39. {
40. Name (_ADR, 0x00080000) // _ADR: Address
```

```

41. Device (B001)
42. {
43. Name (_ADR, Zero) // _ADR: Address
44. Device (C001)
45. {
46. Name (_ADR, 0x00080000) // _ADR: Address
47. Device (D001)
48. {
49. Name (_ADR, Zero) // _ADR: Address
50. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
51. {
52. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
53. {
54. Local0 = Package (0x08)
55. {
56. "AAPL,slot-name",
57. Buffer (0x07)
58. {
59. "Slot-2"
60. },
61.
62. "built-in",
63. Buffer (One)
64. {
65. 0x00 // .
66. },
67.
68. "model",
69. Buffer (0x14)
70. {
71. "Sonnet Fusion SSD 1"
72. },
73.
74. "device_type",
75. Buffer (0x17)
76. {
77. "NVM Express-Controller"
78. }
79. }
80. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
81. Return (Local0)
82. }

```

```
83.
84. Return (Zero)
85. }
86. }
87. }
88.
89. Device (C002)
90. {
91. Name (_ADR, 0x00090000) // _ADR: Address
92. Device (D002)
93. {
94. Name (_ADR, Zero) // _ADR: Address
95. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
96. {
97. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
98. {
99. Local0 = Package (0x08)
100. {
101. "AAPL,slot-name",
102. Buffer (0x07)
103. {
104. "Slot-2"
105. },
106.
107. "built-in",
108. Buffer (One)
109. {
110. 0x00 // .
111. },
112.
113. "model",
114. Buffer (0x14)
115. {
116. "Sonnet Fusion SSD 2"
117. },
118.
119. "device_type",
120. Buffer (0x17)
121. {
122. "NVM Express-Controller"
123. }
124. }
```

```
125. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
126. Return (Local0)
127. }
128.
129. Return (Zero)
130. }
131. }
132. }
133.
134. Device (C003)
135. {
136. Name (_ADR, 0x000A0000) // _ADR: Address
137. Device (D003)
138. {
139. Name (_ADR, Zero) // _ADR: Address
140. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
141. {
142. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
143. {
144. Local0 = Package (0x08)
145. {
146. "AAPL,slot-name",
147. Buffer (0x07)
148. {
149. "Slot-2"
150. },
151.
152. "built-in",
153. Buffer (One)
154. {
155. 0x00 // .
156. },
157.
158. "model",
159. Buffer (0x14)
160. {
161. "Sonnet Fusion SSD 3"
162. },
163.
164. "device_type",
165. Buffer (0x17)
166. {
```

```

167. "NVM Express-Controller"
168. }
169. }
170. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
171. Return (Local0)
172. }
173.
174. Return (Zero)
175. }
176. }
177. }
178.
179. Device (C004)
180. {
181. Name (_ADR, 0x000B0000) // _ADR: Address
182. Device (D004)
183. {
184. Name (_ADR, Zero) // _ADR: Address
185. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
186. {
187. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
188. {
189. Local0 = Package (0x08)
190. {
191. "AAPL,slot-name",
192. Buffer (0x07)
193. {
194. "Slot-2"
195. },
196.
197. "built-in",
198. Buffer (One)
199. {
200. 0x00 // .
201. },
202.
203. "model",
204. Buffer (0x14)
205. {
206. "Sonnet Fusion SSD 4"
207. },
208.

```

```
209. "device_type",
210. Buffer (0x17)
211. {
212. "NVM Express-Controller"
213. }
214. }
215. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
216. Return (Local0)
217. }
218.
219. Return (Zero)
220. }
221. }
222. }
223. }
224. }
225. }
226. }
227. }
```

Alles anzeigen

Beitrag von „pgr69“ vom 1. November 2020, 21:55

[apfelnico](#)

Genau so hat es geklappt. Danke dir

Ich habe zu der SSDT noch eine Frage aber die stelle ich dir morgen ich will heute nicht mehr.

vielen, vielen Dank

Beitrag von „apfelnico“ vom 1. November 2020, 22:25

[pgr69](#)

OK, bis morgen.

Edit: Wahrscheinlich kann man den ganzen Krempel auch in eine Kext packen (Properties Injection). Hätte den Vorteil, falls dieses Problem auch richtige Macs (MacPro) betrifft, könnten diese das unkompliziert nutzen. Denn eine SSDT können die nicht nutzen, es sei denn, die hauen zum Beispiel OpenCore davor. 😊

Aber dennoch interessant, das mal an einer SSDT durch zu exerzieren. Über OpenCore direkt geht das natürlich auch.

Beitrag von „pgr69“ vom 2. November 2020, 11:43

Guten Morgen Miteinander.

[apfelnico](#)

Vielen Dank für Deine Hilfe. Ich habe gestern mehr dazu gelernt als in meiner ganzen Hackintosh-Zeit - zumindest kommt es mir so vor.

Vor Deiner letzten Lösung dachte ich mir schon das die unbekanntes Geräte nacheinander deklariert werden müssen nur allerdings dachte ich das erst eins fertig definiert und injiziert sein muss bevor ich an ein darunterliegendes komme. Dass das so schön kaskadiert in einem rutsch geht ist natürlich klasse.

Gestern stellte sich mir die Frage warum bei _ADR einmal was drin steht und dann wieder nichts.....aber es ist ja nicht NICHTS sondern Zero also 0 und diese Werte kommen von dem was jeweils hinter dem "@" stand oder ?

Bei der Benamung der Devices hast du einfach ABCD genommen und wäre es tiefer gegangen wäre es alphabetisch weiter gegangen mit E usw. ?

Mir ist im IORegExplorer aufgefallen dass da noch andere Devices rumlungern die nur mit ihrer VendorID und DeviceID aufgeführt sind. Weis nur nicht mehr ob es auf dem Sage10g oder auf meinem derzeitigen Designare EX war. Egal....die Frage ist.....diese Devices kann ich doch dann auf die gleiche Art und Weise mit vernünftigen Properties versehen aus denen hervorgeht um was es sich dabei handelt.

Ig und Danke

Beitrag von „apfelnico“ vom 2. November 2020, 16:47

[Zitat von pgr69](#)

Gestern stellte sich mir die Frage warum bei _ADR einmal was drin steht und dann wieder nichts.....aber es ist ja nicht NICHTS sondern Zero also 0 und diese Werte kommen von dem was jeweils hinter dem "@" stand oder ?

Exakt.

[Zitat von pgr69](#)

Bei der Benamung der Devices hast du einfach ABCD genommen und wäre es tiefer gegangen wäre es alphabetisch weiter gegangen mit E usw. ?

Genau. Da es so viele sind und ich nix kreatives mehr leisten wollte, eben so. Was da steht, ist letztendlich (oft) Wurscht. Es müssen max. vier Zeichen sein, Buchstaben und Zahlen sind erlaubt, Zahlen allein gehen nicht. Manche Treiber, Kexte reagieren aber auf bestimmte Namen der ACPI-Devices.

[Zitat von pgr69](#)

Mir ist im IORegExplorer aufgefallen dass da noch andere Devices rumlungern die nur mit ihrer VendorID und DeviceID aufgeführt sind. Weis nur nicht mehr ob es auf dem Sage10g oder auf meinem derzeitigen Designare EX war. Egal....die Frage ist.....diese Devices kann ich doch dann auf die gleiche Art und Weise mit vernünftigen Properties

versehen aus denen hervorgeht um was es sich dabei handelt.

Kann man machen, muss man nicht. Und immer einen Blick in die DSDT werfen. Wenn dort schon eine `_DSM`-Methode vorkommt – bei den vielen Basis-Geräten ist das der Fall – dann geht das so über eine zusätzlich SSDT nicht. Denn innerhalb eines Devices bzw dessen Scope dürfen nicht gleiche Methoden angewendet werden, dann wird diese SSDT ignoriert. In vielen Fällen ist es einfacher, Device Properties im Bootloader festzulegen. Der Vorteil einer SSDT ist aber, dass hier schon auf ACPI-Ebene Dinge repariert und oder verbessert werden, auch lassen sich hier Properties verschiedenen Typs einsetzen. Zum Zeitpunkt in dem der Bootloader dies macht, ist schon eine Menge gelaufen und es können nur nach Properties mit dem Type "DATA" injiziert werden. Oftmals, gerade für die kosmetischen Dinge, ausreichend. Mitunter möchte man aber "Number" nutzen, das geht dann direkt über SSDT.

Beitrag von „pgr69“ vom 2. November 2020, 17:13

[Zitat von apfelnico](#)

Wenn dort schon eine `_DSM`-Methode vorkommt –

Okay dort kann man dann nichts mehr tun weil die `_DSM` schon für diese Geräte abgearbeitet wurde?

Du sagtest wenn die Karte sich im MacPro genauso darstellt wäre für die Leute ein KEXT interessant. Wie geht das? Kann man aus einer SSDT ein Kext machen oder ist die Syntax beim erstellen eines KEXT eine ganz andere?

Ig und Danke

Beitrag von „apfelnico“ vom 2. November 2020, 17:38

Völlig anders. Nur die Grundidee bleibt, einem spezifischen Gerät weitere Properties einzuimpfen.