

BT Modul Instant Wake trotz Definition als 255 (Internal)

Beitrag von „G.com“ vom 30. August 2023, 11:50

Moin,

ich benötige hier einmal Hilfe.

Nachdem ich mich ja nun doch verjüngt habe...stoße ich auf das Problem, dass mein neuer Hack immer einen Instant Wake hat.

USB - Mapping ist sauber gemacht. Ich hatte zunächst den Verdacht, dass das Problem an der noch auf dem Board verbauten Realtek Wifi/BT Lösung liegen kann. Diese ist per M2 gesteckt, aber über HS13/14 im System integriert. Technisch verstehe ich da zwar nur die Hälfte, aber egal.

Nun habe ich Probe halber also auch diese beiden Ports mal intern deklariert, dennoch ist das Problem nicht gelöst. Wake Reason ist folgendes:

Code

1. `mset -g log | grep -e "Sleep.*due to" -e "Wake.*due to"`
- 2.
3. Wake from Normal Sleep [CDNVA] : due to PWRB RP04 RP05/UserActivity Assertion Using AC (Charge:0%)

Mittels `ssdt-gprw` habe ich das Problem zwar letztendlich gelöst verliere aber dadurch die Möglichkeit per USB einen Wake auszulösen.

Das Erfreuliche ist, er schläft ruhig und fest, wie erwartet - ich habe auch keine Abstürze mehr

nach Wake und PowerNap ist sogar funktional.

Habt Ihr Lösungsansätze trotz SSDT-GPRW Wake per Keypress zu aktivieren oder geht es nur bei den Power Button?

Danke für jeden Input.

Viele Grüße

g.com

Beitrag von „KungfuMarek“ vom 30. August 2023, 12:39

Habe auch Probleme mit dem Sleep... das liegt an den Gigabyte Mainboards.

Ist ein allgemeines Problem mit Z690/790 Mainboards von Gigabyte.

Habe gefühlt alles probiert, sleep geht nicht, habe mich damit abgefunden.

Beitrag von „G.com“ vom 30. August 2023, 13:31

[KungfuMarek](#) Also ich widerspreche nur ungerne. Sleep funktioniert astrein. Nur eben durch die SSDT GPRW halt keinen wake per Tastendruck.

Steht oben auch ganz klar geschrieben.

Ich suche einen Weg Wake per Keyboard/Mouse zu aktivieren trotz der SSDT.

Beitrag von „KungfuMarek“ vom 30. August 2023, 13:44

Mit einem älteren BIOS F8 funktioniert Sleep auch, aber dann gehen keine neuen CPUs 13/14 Generation. Deswegen sage ich es geht nicht mehr.

Probiere mal zwei Sleeps hintereinander, dass wird nicht gehen...

Beitrag von „G.com“ vom 30. August 2023, 15:56

[KungfuMarek](#) Moin, auch hier widerspreche ich Dir. Der Second Sleep Bug ist auch gelöst. Das geht mittels boot-arg npci=0x2000.

Bei meinem Board gibt es kein altes Bios ohne 13er Support, wäre ja auch sinnlos bei meinem System.

Ich schaue mir gerne mal deine EFI an und helfe Dir beim Sleep, aber hier würde ich jetzt gerne dabei bleiben meine Frage zu klären.

Beitrag von „KungfuMarek“ vom 30. August 2023, 16:22

Ich habs gerade mal eingepflegt und die SSDT-GPRW.

Ich kann es nicht fassen, es geht! DANKE!

Mir fiel gerade was ein. Hast du diesen Kext mal probiert?

<https://github.com/osy/USBWakeFixup>

Beitrag von „G.com“ vom 30. August 2023, 16:50

[KungfuMarek](#) Jein...ich habe den acpi-wakre-type mittels PI integriert. Leider hat das nichts gebracht. Ich schaue mir das aber noch einmal genauer an.

Fein dass es mit dem Sleep funktioniert.

Zu dem Boot Arg - ich kann nicht genau sagen, was es macht. Es umgeht irgendwelche PCI Routinen und ggf. eliminiert es das im Bios angeschaltete Above 4G - soweit habe ich das noch nicht verstanden. Dies nur zur Info.

Wenn einer genauere Kenntnis hat, was npci=0x2000 macht und ob 4G erhalten bleibt oder nicht - bitte gerne mal Info.

Spoiler anzeigen

UPDATE: Auch der USBWakeupFix Kext bringt keine Möglichkeit das Gerät mittels Keyboard Mouse aufzuwecken.

[al6042](#) [griven](#) Sorry, dass ich Euch direkt anticke, aber Ihr seid doch Spezialisten im Thema DSDT/SSDT und Patching. Habt Ihr da vielleicht einen Tip? Könnt Ihr mir sagen, was PR04/PR05 ist oder wie man die GRPW Methode weniger invasiv anwenden kann, damit Wake per Tastendruck funktioniert?

Beitrag von „al6042“ vom 30. August 2023, 21:36

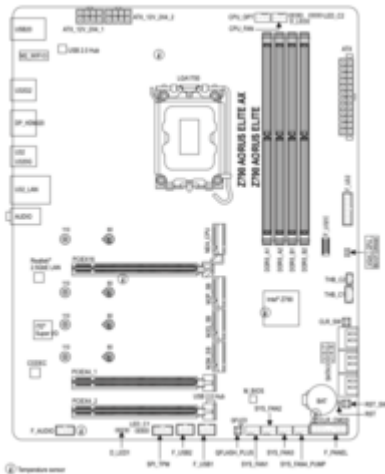
Sorry,

ich hatte die GRPW SSDT in meinem vorherigen Z390, bzw. dem aktuellen Z690 Board nicht mehr im Einsatz und müsste mich erst wieder rein arbeiten um tatsächlich helfen zu können.

Beitrag von „G.com“ vom 30. August 2023, 22:22

Danke Dir [al6042](#). Ich bin nun sogar etwas weiter. Also, das Problem ist tatsächlich meine BT Karte. Habe diese eben mal vom USB abgesteckt und siehe da ohne GPRW schläft er.

Nun ist das ganze aber insofern komplizierter, da der HS Port eigentlich als Internal definiert ist. Leider ist hier die Komplikation, daß beide USB Ports auf dem MB über einen Hub laufen und somit am selben HS Port liegen. Ob dieser Umstand den Fehler verursacht? Schicke gleich einmal Bilder.



Ich habe da zwei Ports F_USB1 und F_USB2.

```
#-----#
#           Discovered USB Ports           #
#-----#
----- USBData Enumeration -----
1. HS01 | AppleUSB200HCIPort | 1 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
2. HS02 | AppleUSB200HCIPort | 2 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
3. HS03 | AppleUSB200HCIPort | 3 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
4. HS04 | AppleUSB200HCIPort | 4 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
5. HS05 | AppleUSB200HCIPort | 5 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
6. HS06 | AppleUSB200HCIPort | 6 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
7. HS07 | AppleUSB200HCIPort | 7 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
8. HS08 | AppleUSB200HCIPort | 8 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
9. HS09 | AppleUSB200HCIPort | 9 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
10. HS10 | AppleUSB200HCIPort | 10 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
11. HS11 | AppleUSB200HCIPort | 11 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
12. HS12 | AppleUSB200HCIPort | 12 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
13. HS13 | AppleUSB200HCIPort | 13 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
14. HS14 | AppleUSB200HCIPort | 14 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
15. HS15 | AppleUSB200HCIPort | 15 | 00000000 | 04000000 | Type 0
USB 2.0 Nopop
Populated:
USB 2.0
```

Beide sind in einem HUB zusammengefasst.

| Type | Name | Location ID | Port | Connector | Max Speed | Device | Comment |
|------|------|-------------|------|-----------|-----------|--------------------|---------------|
| HKCD | HS01 | 0x18300000 | 0x01 | Type-C-40 | 480 Mbps | Linkstation | USB-C Front |
| HKCD | HS02 | 0x18300000 | 0x02 | Type-C-40 | 480 Mbps | Linkstation | USB-C Front |
| HKCD | HS06 | 0x18300000 | 0x06 | USB-A | 480 Mbps | Backlogic Hub | USB2 Front |
| HKCD | HS06 | 0x18400000 | 0x06 | USB-A | 480 Mbps | Linkstation | USB2 Front |
| HKCD | HS07 | 0x18300000 | 0x07 | USB-A | 480 Mbps | Linkstation | USB2 Front |
| HKCD | HS09 | 0x18300000 | 0x09 | USB-A | 480 Mbps | Linkstation | USB2 Front |
| HKCD | HS11 | 0x18700000 | 0x08 | Internal | 480 Mbps | USB2.0 Hub | USB2 Internal |
| HKCD | HS12 | 0x18800000 | 0x0C | Internal | 480 Mbps | USB2.0 Hub | USB2 Internal |
| HKCD | SS03 | 0x18300000 | 0x13 | USB-A | 5 Gbps | External Drive 3.0 | USB2 Front |
| HKCD | SS04 | 0x18300000 | 0x14 | USB-A | 5 Gbps | Backlogic Hub | USB2 Front |
| HKCD | SS06 | 0x18300000 | 0x16 | USB-A | 480 Mbps | Linkstation | USB2.0 Front |
| HKCD | SS07 | 0x18300000 | 0x17 | USB-A | 480 Mbps | Linkstation | USB2.0 Front |
| HKCD | SS08 | 0x18300000 | 0x18 | USB-A | 480 Mbps | Linkstation | USB2 Front |
| HKCD | SS09 | 0x18300000 | 0x19 | USB-A | 480 Mbps | Linkstation | USB2 Front |

Definiert, als Internal.

Dennoch habe ich den Instant Wake vom BT Modul. Dieses Problem bestand zu keinem Zeitpunkt unter meiner alten Maschine.

Was ist da los? 😊

Beitrag von „griven“ vom 30. August 2023, 22:55

Interessant ist das die Tastatur/Mouse bzw. deren Empfänger auch an dem Hub hängen...

Ich könnte mir vorstellen das sich die BT Karte und der Empfänger an dem als Intern

definierten HUB in die Quere kommen.

Beitrag von „G.com“ vom 30. August 2023, 23:06

Diese Vermutung hatte ich auch [griven](#), aber selbst das Abstöpseln des Front USB und das Umstecken des Empfängers hat leider keine Besserung gebracht.

Ich bin hier gerade mit dem Latein am Ende.

Versuche jetzt mal mit USB Reset - bezweifle aber, das es etwas bringt.

UPDATE: Wie vermutet, die Einbindung der RHUB bringt keine Verbesserung. Ich würde mir wünschen mit Eurer Hilfe hier eine Lösung zu finden, da mein Ziel ja eigentlich war ein sauberes Rapture Lake Tutorial nebst passender EFI hier zu posten. Probleme sind ja nur noch ganz wenige da. Aber, Ihr wisst ja, man will immer das Maximum.

Was hat nicht sauber läuft sind 4 Punkte. Dies ist einer davon. Das DRM Problem ist ein Kabel/HDCP Problem, funktioniert aber. BT Entsperren mit der Watch könnte an der Hardware oder an der SSDT-GPRW liegen und dann gibt es nur noch eine Lappalie.

Danke für die Unterstützung.

Beitrag von „griven“ vom 30. August 2023, 23:39

Entsperren mit der Watch ist generell nicht das zuverlässigste unter der Sonne selbst an originaler Apple Hardware nicht...

Bei meinem M1 MacBookPro funktioniert das auch nur wenn es gerade Lust dazu hat. Sehr häufig geht es halt auch nicht und behauptet dann immer die drahtlose Verbindung sei zu

schwach *BlaaBluub* so richtig ausgegoren ist das demnach offenbar alles nicht (in Meinem Fall eine AppleWatch Series 6 mit watchOS10 und eben das M1 mit Sonoma). Was die USB Probleme bei Dir angeht sollte Sleep bei korrekt als intern definierten Ports das BT Modul doch eigentlich gar nicht kümmern weil dem wird ja dann im Sleep der Saft nicht abgedreht 🤔

Beitrag von „G.com“ vom 30. August 2023, 23:46

Eigentlich ja und auf der alten Kiste hat das BT auch bei intern definertem USB nicht den Sleep gestört.

Hier in dem neuen System hingegen will es ohne die Brechstange _PRW zu kappen nicht funktionieren.

Ich meine, Du siehst ja - die Ports sind sauber definiert. Im Bios habe ich ERP an/aus und ASPM an/aus probiert ändert nichts.

Ich kann gerne mal die Efi posten, vielleicht habe ich etwas übersehen.

So richtig schlau werde ich aber nicht daraus.

Wie im anderen Post erwähnt hatte ich ja das Realtek Modul in Verdacht, dies ist über Hs 13 und HS 14 eingebunden, aber selbst das als intern definiert macht keinen Unterschied. Letztendlich der Beweis, dass es der BT USB Anschluss ist, kam dann durch abziehen des Kabels. Bäm, er schläft. Kabel dran Instant Wake.

Jetzt könnte ich natürlich die Karte vom PCI Adapter abziehen und in den Socket vom MB packen. Verliere dann aber auch das AX Modul unter Windows. Letztlich ist der Rechner ja eh über LAN am 2,5gbit Switch. Aber ob das mein Problem löst? Und vor allem ob es dann mit den Antenne. Anschlüssen von der Realtek Karte harmoniert?

Alternativ könnte ich natürlich eine Fenwi ordern und testen, aber ich vermute das gleiche Problem in grün.

UPDATE:

Ich habe jetzt mit acpi-wake-types experimentiert, habe die pci-aspm-defaults für den HXCI Controller eingerichtet. mit einem XHCI-Z790-unsupported.kext experimentiert. Problem bleibt bestehen. Instant Wake solange der blöde Dengele am USB hängt.

Meine EFI anbei - ich bin wirklich offen für Unterstützung -BTW - die Serial ist frisch generiert.

[griven](#) Es wird spannend. Also, nach langer Recherche stieß ich auf die Wurzel des Übels. Per USB mapping definiert man zwar den übergeordneten HS Port (das Hub) aber nicht die darunter liegenden Subports. Diese werde einfach nicht als intern wahrgenommen ergo...instant Wake. Es wäre also ein Ansatz entweder dem/denen die Methode _PRW zu entziehen per SSDT oder per Property Injection Werte zuzuweisen, die sie als interne Ports definieren. Jetzt bin ich gerade fern vom Rechner sehe hier aber meine Möglichkeit. Wenn Ihr da einen Ansatz seht, bitte gerne mal hier hinterlassen.

Lese ich mir nun die SSDT Methode von [apfelnico](#) durch werden die im genannten Wake Reson RP4&RP5 die beiden Hubs sein. Frage wäre also, wie ich diese dann sauber deklariere, damit es bis tief in die Subports greift.

Ok, wer lesen kann ist klar im Vorteil. Die SSDT Methode ist klar ein Ansatz, der RP und die dazugehörigen Unterports werden da sauber deklariert. Könnte gut sein, dass ich hier Hilfe benötige, da meine SSDT von den Methoden ganz anders aufbaut. Habe es deswegen zunächst verworfen.

Ich danke schon jetzt für jede Hilfe.

Beitrag von „griven“ vom 1. September 2023, 08:39

Puh ich bin bei dem Thema eher raus glaube ich denn um das USB Zeug habe ich bisher immer einen sehr sehr weiten Bogen geschlagen und mich mit dem begnügt was aus der ToolBox gefallen ist (zugegebenermaßen hatte ich bisher auch nicht die Notwendigkeit mich damit zu beschäftigen weil es halt einfach funktioniert hat Glück schätze ich). Was die SSDT Methode für USB Deklarationen angeht meine ich mich aber zu erinnern das der [kaneske](#) da seinerzeit recht aktiv mit dabei war und daher wohl in die Richtung vermutlich einiges Wissen parat hat. Deine Beobachtung mit den Ports könnte aber durchaus passen sprich das sich das Property nicht auf die untergeordneten Ports vererbt...

Früher oder später werde ich da aber wohl oder übel auch ran müssen denn das Elitebook will mit Sonoma auch nicht so recht im Sleep bleiben unter Ventura kein Problem sprich da werde ich auch suchen müssen woran das liegt...

Beitrag von „kaneske“ vom 1. September 2023, 09:37

[G.com](#) ~~uppe mal bitte die Stock SSDT A M I oder wie auch immer deine USB SSDT heißt zusätzlich zu deiner bearbeiteten.~~ Dann guck ich da auch mal drauf. Aber dein Ansatz wird richtig sein, du musst bis zum Port runter. Das ist das Blöde an diesen Hubs.

Ok zu blöd meinerseits, lag ja in der EFI.

Hab mir das mal angeschaut, ja weicht erheblich von dem was ich bisher gesehen habe ab.

In der SSDT-11.aml finden sich deine USB-Ports wieder, Table Länge: 0x00001651 (5713)

Weiter werden bei dir die Ports anhand der Variablen PU2C und PU3C aktiviert, bzw die Properes die in der SSDT deklariert sind aktiviert. z.B. If ((0x02 <= PU3C)) wird nur aktiv bei PU3C == 2 oder mehr...

Code

1. If ((One <= PU2C))

somit limitiert sich das hier schon mal, was aber nicht schlimm ist, das kannst du ja so stehen lassen für deine Methoden.

Kannst du ja sehen, betrifft wiederum dann den Port HS01 direkt, den kannst du wie Nico es beschrieb bearbeiten, was dann so aussehen kann:

Code

1. If ((One <= PU2C))
2. {
3. Scope (_SB.PC00.XHCI.RHUB.HS01)
4. {
5. Method (_STA, 0, NotSerialized) // _STA: Status
6. {
7. Return (0x0F)
8. }
- 9.
10. Name (_UID, One) // _UID: Unique ID
11. Name (_UPC, Package (0x04) // _UPC: USB Port Capabilities
12. {
13. 0xFF,
14. 0xFF,
15. Zero,
16. Zero
17. })
18. Name (_PLD, Package (0x01) // _PLD: Physical Location of Device
19. {
20. ToPLD (
21. PLD_Revision = 0x1,
22. PLD_IgnoreColor = 0x1,
23. PLD_Red = 0x0,
24. PLD_Green = 0x0,

```

25. PLD_Blue = 0x0,
26. PLD_Width = 0x0,
27. PLD_Height = 0x0,
28. PLD_UserVisible = 0x1,
29. PLD_Dock = 0x0,
30. PLD_Lid = 0x0,
31. PLD_Panel = "UNKNOWN",
32. PLD_VerticalPosition = "UPPER",
33. PLD_HorizontalPosition = "LEFT",
34. PLD_Shape = "UNKNOWN",
35. PLD_GroupOrientation = 0x0,
36. PLD_GroupToken = 0x0,
37. PLD_GroupPosition = 0x0,
38. PLD_Bay = 0x0,
39. PLD_Ejectable = 0x0,
40. PLD_EjectRequired = 0x0,
41. PLD_CabinetNumber = 0x0,
42. PLD_CardCageNumber = 0x0,
43. PLD_Reference = 0x0,
44. PLD_Rotation = 0x0,
45. PLD_Order = 0x0,
46. PLD_VerticalOffset = 0x0,
47. PLD_HorizontalOffset = 0x0)
48.
49. })
50. Method (_PRW, 0, NotSerialized) // _PRW: Power Resources for Wake
51. {
52. Return (Package (0x02)
53. {
54. 0x6D,
55. 0x04
56. })
57. }
58.
59. Method (_CRS, 0, Serialized) // _CRS: Current Resource Settings
60. {
61. Name (ABUF, Buffer (0x02)
62. {
63. 0x79, 0x00 // y.
64. })
65. Return (ABUF) /* \_SB_.PC00.XHCI.RHUB.HS13._CRS.ABUF */

```

```

66. }
67.
68. Method (DTGP, 5, NotSerialized)
69. {
70. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
71. {
72. If ((Arg1 == One))
73. {
74. If ((Arg2 == Zero))
75. {
76. Arg4 = Buffer (One)
77. {
78. 0x03 // .
79. }
80. Return (One)
81. }
82.
83. If ((Arg2 == One))
84. {
85. Return (One)
86. }
87. }
88. }
89.
90. Arg4 = Buffer (One)
91. {
92. 0x00 // .
93. }
94. Return (Zero)
95. }
96.
97. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
98. {
99. If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
100. {
101. Local0 = Package (0x08)
102. {
103. "baud",
104. Buffer (0x08)
105. {
106. 0xC0, 0xC6, 0x2D, 0x00, 0x00, 0x00, 0x00, 0x00 // ...-.....

```

```

107. },
108.
109. "parity",
110. Buffer (0x08)
111. {
112. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 // .....
113. },
114.
115. "dataBits",
116. Buffer (0x08)
117. {
118. 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 // .....
119. },
120.
121. "stopBits",
122. Buffer (0x08)
123. {
124. 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 // .....
125. }
126. }
127. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
128. Return (Local0)
129. }
130.
131. Return (Zero)
132. }
133. }

```

Alles anzeigen

(Variante aus seinem Thread in der er HS13 extra für Bluetooth noch erweitert gepatched hat.)

Einfach wäre es dann so:

Code

1. If ((One <= PU2C))
2. {
3. Scope (_SB.PC00.XHCI.RHUB.HS01)
4. {

```
5. Name (_UPC, Package (0x04) // _UPC: USB Port Capabilities
6. {
7. 0xFF,
8. 0xFF,
9. Zero,
10. Zero
11. })
12. Name (_PLD, Package (0x01) // _PLD: Physical Location of Device
13. {
14. ToPLD (
15. PLD_Revision = 0x1,
16. PLD_IgnoreColor = 0x1,
17. PLD_Red = 0x0,
18. PLD_Green = 0x0,
19. PLD_Blue = 0x0,
20. PLD_Width = 0x0,
21. PLD_Height = 0x0,
22. PLD_UserVisible = 0x1,
23. PLD_Dock = 0x0,
24. PLD_Lid = 0x0,
25. PLD_Panel = "UNKNOWN",
26. PLD_VerticalPosition = "UPPER",
27. PLD_HorizontalPosition = "LEFT",
28. PLD_Shape = "UNKNOWN",
29. PLD_GroupOrientation = 0x0,
30. PLD_GroupToken = 0x0,
31. PLD_GroupPosition = 0x0,
32. PLD_Bay = 0x0,
33. PLD_Ejectable = 0x0,
34. PLD_EjectRequired = 0x0,
35. PLD_CabinetNumber = 0x0,
36. PLD_CardCageNumber = 0x0,
37. PLD_Reference = 0x0,
38. PLD_Rotation = 0x0,
39. PLD_Order = 0x0,
40. PLD_VerticalOffset = 0x0,
41. PLD_HorizontalOffset = 0x0)
42.
43. })
44.
45. }
46.
```

47. }

Alles anzeigen

Könnte das dann so aussehen...schau aber bitte noch mal genau drüber ich bin da auch nicht Nico was das angeht.

Beitrag von „G.com“ vom 2. September 2023, 05:00

[kaneske](#) Ich habe bis gerade eben schon experimentiert, allerdings deine Anmerkungen noch nicht gesehen.

Hatte es aber genau so gemacht.

Zu der Feststellung von Dir kommt erschwerend hinzu, dass die PR Ports vom XHCI dann in der SSDT-9 definiert werden. Wiederum mit eine krassen Anzahl von Methoden.

Ich bin hier total lost. Meine bearbeiteten SSDT's enden in einer Menge ACPI Fehlern, somit greifen die nicht. Das wäre aber sicher zu korrigieren.

Des Weiteren sehe ich im IOReg, dass die PR Ports aus der SSDT-9 gar nicht die ausschließlich die USB Ports zu bedienen scheinen. Ein Beispiel sei hier der PR09 oder PR25. Sieht dann so im IOReg aus.

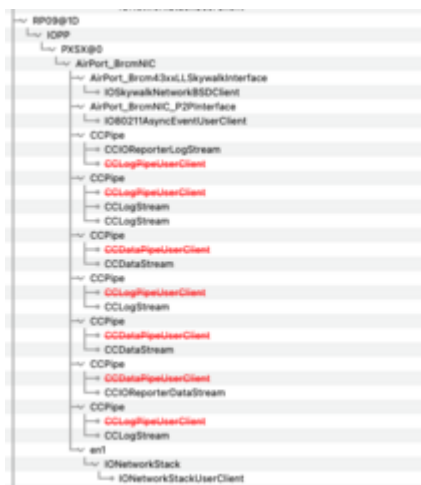


Da hängt mal geschmeidig die NVME dran.

```
}
}
} (IOAPIC == 0x1) && (IOAPIC == 0x1) || (IOAPIC == 0x1) || (IOAPIC == 0x1)
{
  Name (_SB.PCI0.FXSB.FXSB0)
  Name (TMRP, 0x1)
  Device (TMRP)
  {
    Name (_ADR, 0x00000000) // _ADR: Address
    Device (MWB1)
    {
      Name (_ADR, 0x00) // _ADR: Address
      Device (MWB0)
      {
        Name (_ADR, 0x00) // _ADR: Address
        Method (MWB0, 1, SerialIntr)
        {
          Local0 = (TMRP == 0x1)
          Local1 = 0x0
          if ((TP12 == 0x1) == Local0) && (Intr == TP12)
          {
            Return (TP12 & 0x1)
          }
          Case0: (|(TP20 == 0x1) == Local0) && (Intr == TP20)
          {
            Return (TP20 & 0x1)
          }
          Case1: (|(TP30 == 0x1) == Local0) && (Intr == TP30)
          {
            Return (TP30 & 0x1)
          }
          Case2: (|(TP40 == 0x1) == Local0) && (Intr == TP40)
          {
            Return (TP40 & 0x1)
          }
          Case3: (|(TP50 == 0x1) == Local0) && (Intr == TP50)
          {
            Return (TP50 & 0x1)
          }
          Case4: (|(TP60 == 0x1) == Local0) && (Intr == TP60)
          {
            Return (TP60 & 0x1)
          }
        }
      }
    }
  }
}
```

Und der hintere USB 2 Hub wiederum taucht im IOReg unter keinem RP auf, auch RP09 ist der Wifi Anteil der Broadcom Karte im PCI Steckplatz.

Warum die alle über RHUB bzw. XHCI eingebunden werde ist mir absolut unklar. Oder gleiche Namen für unterschiedliche Dinge...



Ich denke das wird ein steifes Projekt und ggf. kann ich das mit Eurer Hilfe schaffen.

Ich wäre Dir massiv dankbar, wenn Du mir hier weiter Support geben könntest.

Ich denke das Ergebnis könnte hier für das Forum auch für viele andere interessant sein, da es wohl möglicherweise etliche AMI Biosse mit ähnlichen Strukturen geben mag.

Jetzt fehlt mir allerdings leider erst einmal das richtige Kabel um die USB-C Ports mit Full Speed zu identifizieren. Da muss ich bis Montag warten.

Ich werde morgen schon einmal vorarbeiten und melde mich dann.

UPDATE 1:

Was mich stutzig macht. Wenn der Hauptport als 255 deklariert wird, dann schalten die Apple Treiber die Unterports auf Port-Type 2, Ist der Hauptport als USB2 deklariert, werden die Port-Typen der darunter liegenden Ports auf Port-Type 0 geschaltet.

Das Verhalten ist reproduzierbar. Ganz gleich, ob mit Kext oder per SSDT. Ist das

wenn Du noch eine Idee hast, wäre ich dankbar.

Für den Moment sehe ich 3 Möglichkeiten. Mit _GPRW und kein Wake on USB leben, Karte in das interne M2 Fach einbauen, über meine USB3 Controller PCI Karte und Adapter einbinden.

Ich erinnere mich, dass wir von 15 Ports pro Controller sprechen, also hätte das den Nebeneffekt, ich habe zusätzliche 3 USB3,1 Ports.

Beitrag von „N0b0dy“ vom 3. September 2023, 13:38

Lade bitte mal die original ACPI hier hoch oder schick sie mir PN mit einer gespeicherte Kopie von iOReg.

ACPI kannst du am besten mit Clover bootlader ziehen

Beitrag von „kaneske“ vom 3. September 2023, 14:47

Seine Origin liegt in der EFI oben N0b0dy

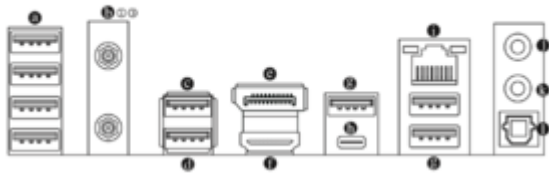
Beitrag von „N0b0dy“ vom 3. September 2023, 19:01

Ich habe die hinterlegte USBPorts.kext in der EFI angeschaut und gehe davon, dass die USB-Mapping ist nicht richtig sauber...!

Ich kenne von vielen MBs, wenn ein Port USB2.0 und USB3.0 unterstützt dann haben die oft die gleiche Nummer aber von Gigabyte weiß ich nicht ob die anderes sind als von Asus oder MSI

Daher soll [G.com](#) uns verraten, wie er USB-Mapping gemacht hat und welche Nummer zu welchem Port gehört..!

2-6 Back Panel Connectors



Beitrag von „G.com“ vom 3. September 2023, 21:04

N0b0dy Ich werde noch einmal alles auflisten. Danke für das Hilfsangebot. Ich habe das USB Mapping mit USBToolbox unter Windows gemacht und danach dann noch einmal den Kext mit Hackintool erstellt.

In MacOS habe ich dazu die RHUB-reset und den XHCI Port Limit Quirk benutzt.

Ein IOReg Dump von dem Zustand, also alle Ports sichtbar habe ich mir erstellt.

Ganz klassisch dann mit USB 2 und USB 3 Gerät Port für Port geprüft und die Port Adresse vermerkt, um dann von Hex auf den Wert zu gehen.

Was mir neu ist, die HS Ports laufen unter 14x... die SS unter 15x... Zumindest ohne Patching.

Und ja, bei Gigabyte ist es leider nicht so, das HS03 mit SS03 verbunden ist. Aber das werde ich noch einmal eruieren. Bzw. schau auch mal oben auf den Screenshot vom USBMap, da habe ich sehr saubere Comments erfasst. z. B der hintere USB-A 3.1 Port unten unten hat HS05 und SS04.

Ich nehme gerne Eure Hilfe an. Allerdings fehlt es mir an einem Kabel, um die SS Personality der USB-C festzustellen. Habe wohl nur USB2.0 Kabel hier. Ich erwarte da morgen die Amazon Lieferung, bin allerdings beruflich bis Mittwoch verreist. Somit macht es am meisten Sinn mich dann am Ende der Woche zu melden.

Danke noch einmal

Beitrag von „N0b0dy“ vom 4. September 2023, 08:03

Okay cool d.h. Quirk XhciPortLimit funktioniert wieder, dann bin ich nicht mehr auf dem aktuellen Zustand, da ich eignen Weg nutze und habe ihn [hier](#) schon mal beschrieben außerdem kann ich kein SSDT für USBs erstellen, wenn ich die Ports nicht weiß und welche die genutzt werden soll.

Bis dahin gute Reise

Beitrag von „G.com“ vom 6. September 2023, 22:31

Moin N0b0dy,

danke noch einmal für dein Angebot der Hilfe. Ich finde das Ganze hier gerade sehr verzwick.

Aber mal von vorne. Im Grunde finde ich in der DSDT die üblichen 26 Ports:

HS01 - HS014

USR01, USR02

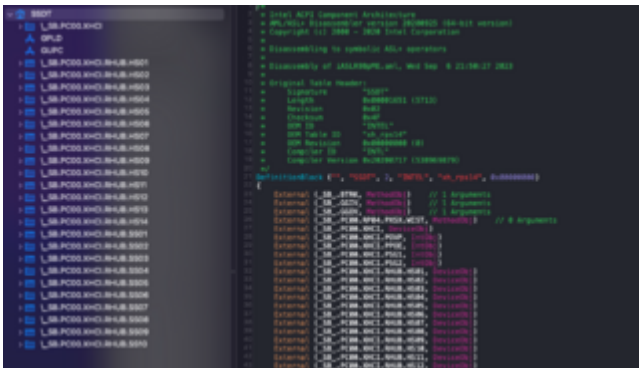
SS01 - SS10



In der SSDT-11 werden dann allerdings "nur" folgende Ports definiert.

HS01-HS14

SS01-SS10



Wenn ich dann die SSDT-USB-Reset und das XHCI Port Limit nutze, finde ich im IOReg und im Mapping Tool dann aber 25 Ports:

16x HS

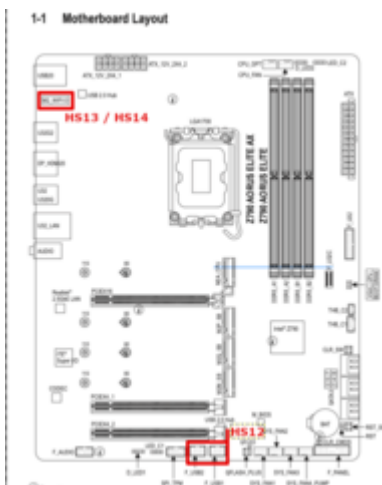
9x SS

| | USB 2 Personality | active | USB 3 Personality | active |
|-----------------------------|-------------------|--------|-------------------|--------|
| USB-C front | HS03 | x | SS02 | x |
| USB 2.0 Hub front/internal | HS12 | x | | |
| USB 3.2 Gen.1 front links | HS09 | | SS08 | x |
| USB 3.2 Gen.1 front rechts | HS10 | | SS09 | x |
| ITE Device | HS13 | | | |
| BT Radio | HS14 | | | |
| USB 2.0 Hub back | HS11 | x | | |
| USB 3.2 Gen.2 hinten oben | HS08 | | SS07 | x |
| USB 3.2 Gen.2 hinten unten | HS07 | | SS06 | x |
| USB 3.2 Gen.1 hinten mittig | HS04 | | SS03 | x |
| USB-C hinten mittig | HS01 | x | SS01 | x |
| USB 3.2 Gen.1 hinten oben | HS06 | x | SS05 | x |
| USB 3.2 Gen.1 hinten unten | HS05 | x | SS04 | x |
| | | 6 | | 9 |
| | | | 15 | |

Ich habe mal vor und nach dem Mappen ein IOReg Dump gemacht, so kannst Du noch einmal selber schauen. Vielleicht habe ich ja einen Knoten im Kopf. PXSX ist meine USB3.0 Karte, die ich vorsorglich eingebaut habe, um evtl. das BT Modul darüber einzubinden. Kann zunächst missachtet werden.

However, dies entspricht dann der folgenden Port Nummerierung.

Internal



Problem hier ist halt, dass der Front USB2.0_1 + _2 beide über ein Hub laufen und eben auf



Folge ich jetzt deiner Aussage zu den Port Nummern und nähme an Adresse 1500000 wäre jetzt der erste USB3 Port (falsch deklariert) dann würden wiederum fast alle SS und HS Ports Nummergleich sein. Aber wenn ich jetzt den Kext ziehe nummeriert er es genau so durch, wie von mir angegeben.

Ich habe allerdings Probleme auf HS03/SS04 - da erkennt er mit dem Kext meine Laufwerke nicht. Ich schließe also nicht aus, dass hier der Wurm drin ist.

Es sei noch einmal kurz dazu gesagt, die PR XHCI Definitionen sind bei mir in der SSDT-09. Allerdings tauchen im IOReg darauf nur PCI Geräte auf.

Zu guter Letzt habe ich noch einmal ein IOReg Dump mit Kext gemacht. So kannst Du auch das noch einmal abgleichen.

Schicke ich Dir dann per PN.

P.S. Hier noch einmal die Fehlermeldung bzw. der Wake Reason:

[Wake from Normal Sleep \[CDNVA\] : due to PWRB RP04 RP05/UserActivity Assertion Using AC \(Charge:0%\)](#)

Zu dem Zeitpunkt steckte die BCMR aber noch in einem anderen Steckplatz und tauchte nicht unter PR05 sondern unter PR09 auf - das steckt jetzt die USB-3 Karte drin.

Hatte ich schon Danke gesagt? Kann man an dieser Stelle sicher nicht oft genug sagen, das hier ist schon echt höhere Mathematik oder ich zu alt.

Viele Grüße

g.com

Beitrag von „N0b0dy“ vom 7. September 2023, 10:59

[Zitat von G.com](#)

Wenn ich dann die SSDT-USB-Reset und das XHCI Port Limit nutze, finde ich im IOReg und im Mapping Tool dann aber 25 Ports

Das ist ja komisch, sie sollten 26 Ports sein

141.... bis 14E... sind HS01 bis HS14

14f und 150 sind dummy USB

151 bis 15a sollten SS01 bis SS10 sein

[Zitat von G.com](#)

Wenn ich also annehme, der erste 3.0 Port sei SS01

Was meinst du da hier? habe nicht verstanden!

15100000 sollte erste SS sein...

[Zitat von G.com](#)

PXSX ist meine USB3.0 Karte, die ich vorsorglich eingebaut habe, um evtl. das BT Modul darüber einzubinden. Kann zunächst missachtet werden

Zwar hier eine externe Karte mit eigenem Controller aber trotzdem müssen die Ports deklariert, außerdem was für eine Karte ist sie?

[Zitat von G.com](#)

Ich habe allerdings Probleme auf HS03/SS04 - da erkennt er mit dem Kext meine Laufwerke nicht. Ich schließe also nicht aus, dass hier der Wurm drin ist.

Ist es hier ein Tippfehler oder so deklariert, sollte hier nicht HS05/SS04 sein, den Port hinten unten wie du sie nennst?

[Zitat von G.com](#)

Es sei noch einmal kurz dazu gesagt, die PR XHCI Definitionen sind bei mir in der SSDT-09. Allerdings tauchen im IOReg darauf nur PCI Geräte auf.

Diese Ports sind mit Thunderbolt verbunden und haben mit USB controller nicht zu tun.

Beitrag von „G.com“ vom 7. September 2023, 11:29

N0b0dy

[Zitat von N0b0dy](#)

Was meinst du da hier? habe nicht verstanden!

15100000 sollte erste SS sein...

Korrekt, so war es gemeint. Ich wollte nur sagen, wenn 15000000 evtl. falsch deklariert wäre...dann. Egal, wir meinen das Gleiche 😊

[Zitat von N0b0dy](#)

Das ist ja komisch, sie sollten 26 Ports sein

141.... bis 14E... sind HS01 bis HS14

14f und 150 sind dummy USB

151 bis 15a sollten SS01 bis SS10 sein

Eben drum, 15a... gibt es eben nicht, wird nicht geladen, obwohl in der SSDT-11 deklariert.

[Zitat von N0b0dy](#)

Ist es hier ein Tippfehler oder so deklariert, sollte hier nicht HS05/SS04 sein, den Port hinten unten wie du sie nennst?

Ja... es war ja gestern schon spät und ich hatte 700km im Rücken, Korrekt HS05/SS04. Tippfehler

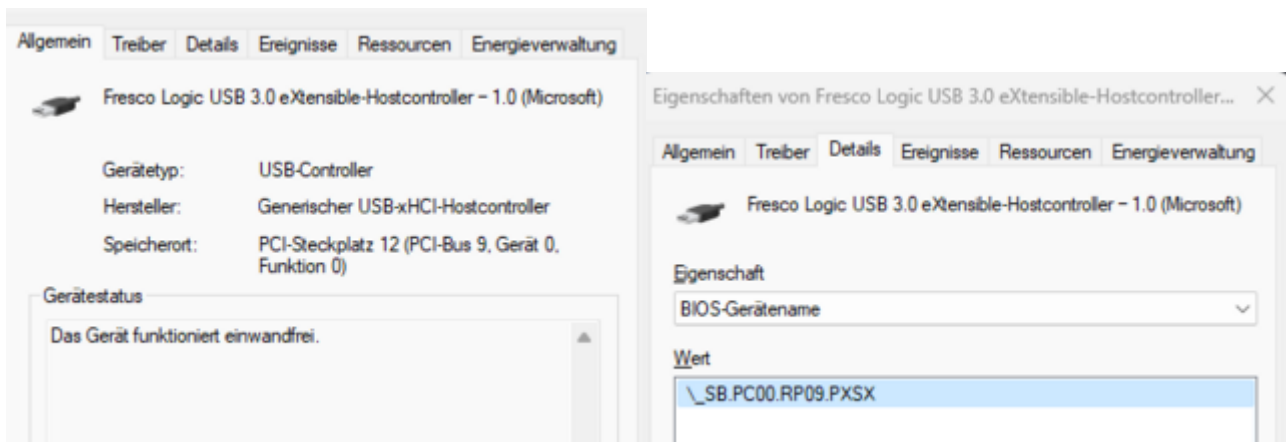
[Zitat von N0b0dy](#)

Diese Ports sind mit Thunderbolt verbunden und haben mit USB controller nicht zu tun.

Danke für die Erklärung, d.h. es gibt keine Definition der Hubs, wie bei Nico beschrieben. Ergo, die Appletreiber entscheiden, wie die darunterliegenden Ports definiert werden. Erfahrung zeigt, wenn ich HS12 auf 255 stelle, werden die Ports im Hub auf Port-Type 0x02 geschaltet. Was immer das genau bedeutet. Stelle ich HS12 auf 0, dann ist der Port Type 0x00. Ich vermute hier das Problem.

Die Definition der USB3 Karte muss noch gemacht werden, außer - wir sind erfolgreich, dann baue ich Sie nämlich aus. Deswegen dachte ich zunächst, man könne diese missachten. Ist eine Fresco 1000.

https://www.amazon.de/gp/produ...tle_o00_s00?ie=UTF8&psc=1



Es sei noch erwähnt. Interessanterweise passiert es, wenn ich die SSDT-11 droppe und sie unverändert über OC lade, tauchen alle SSXX Ports nicht mehr auf. Diese werden nicht initialisiert. Selbst, wenn ich einzelne HS Ports per _STA deaktiviere, dann werden es halt weniger Ports, aber es werden keine SS Ports initialisiert. Soweit hatte ich es schon getestet. Liegt also nicht am Port Limit.

Gruß

g.com