Neuer Versuch USB per SSDT zu deklarieren / Probleme mit dem internen USB Hub

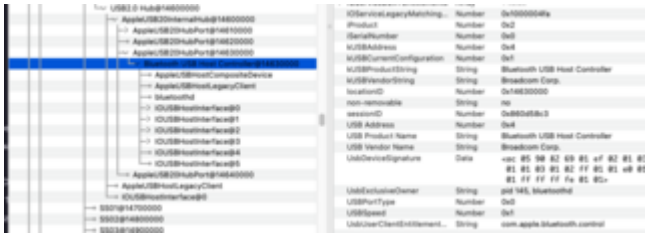
Beitrag von „G.com“ vom 18. November 2024, 21:28

Moin apfelnico

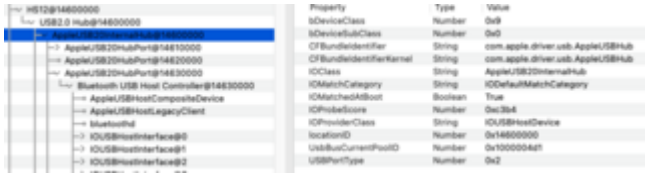
Hier nun der zugesagte Thread.

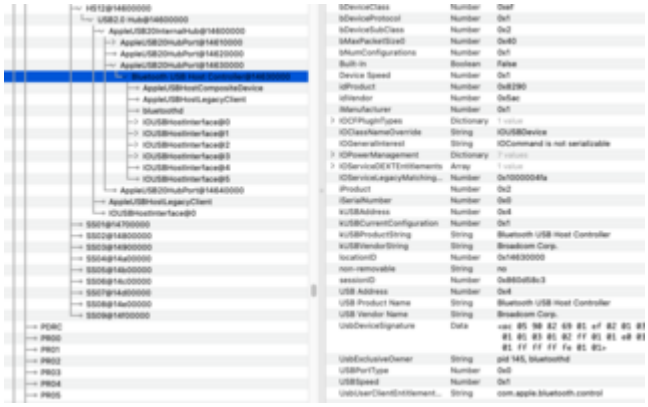
Problem hier ist, dass USB2.0_1 + _2 beide über ein Hub (HS12) laufen. Insgesamt hat das HS12 also 4 Ports.

HS12 ist als intern (255 deklariert):



Egal mit welcher Methode per SSDT oder auch per Kext experimentiert wurde, die darunterlegenden Hubs sind immer 2 oder 0. Aktuell nutze ich daher den Kext.





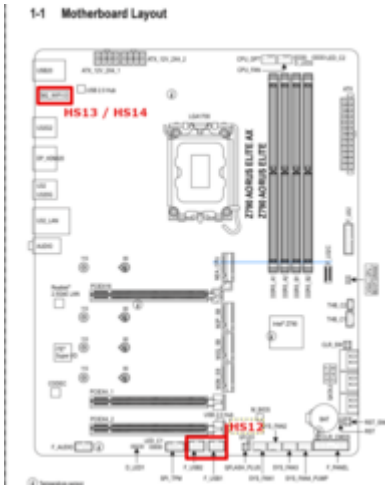
Das führt dann zum Instant Wake Problem und ohne GPRW Patch bekomme ich Ihn nicht in den Sleep.

Die zu erwartende Fehlermeldung bzw. der Wake Reason:

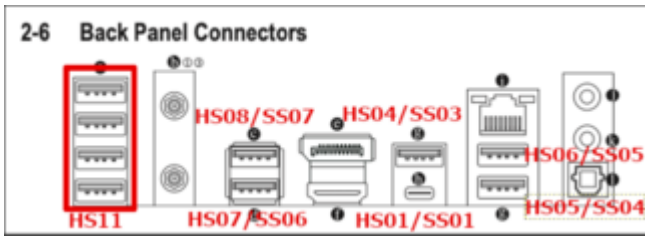
Wake from Normal Sleep [CDNVA] : due to PWRB RP04 RP05/UserActivity Assertion Using AC (Charge:0%)

Hier die notwendigen Infos:

Internal



Backpanel



Front

Die DSDT führt die üblichen 26 Ports auf:

HS01 - HS014

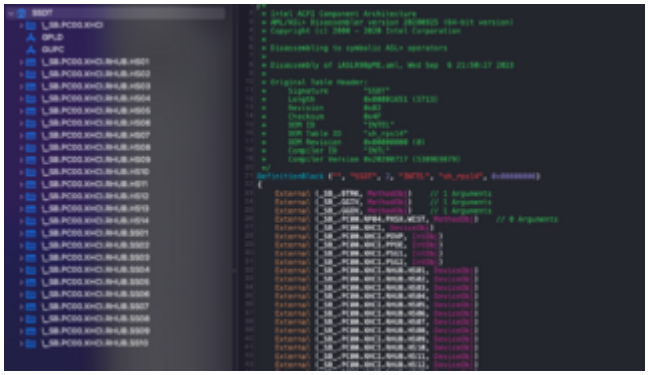
USR01, USR02

SS01 - SS10

In der SSDT-11 werden dann allerdings folgende Ports definiert. Soweit so klar

HS01-HS14

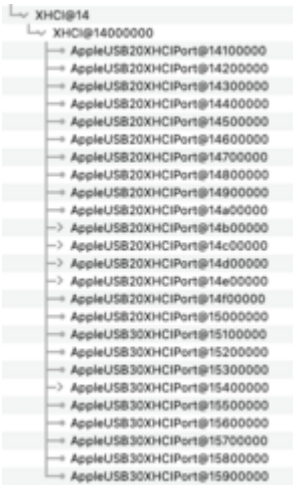
SS01-SS10

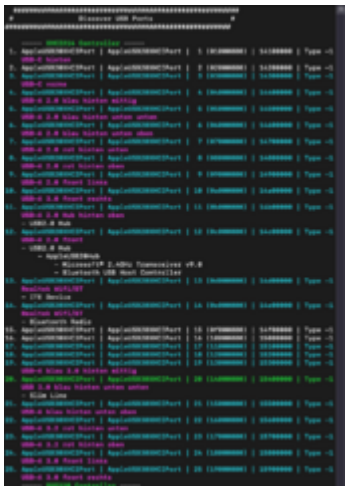


Wenn ich dann die SSDT-USB-Reset und das XHCI Port Limit nutze, finde ich im IOReg und im Mapping Tool dann aber "nur" 25 Ports:

16x HS

9x SS

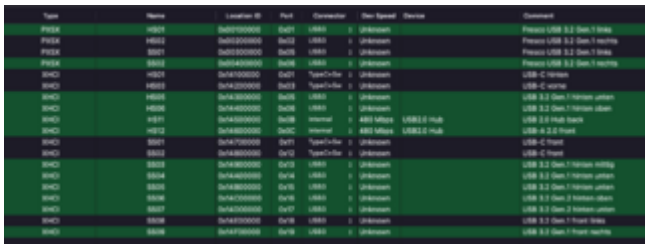




Ich zähle durch und definiere die zu verwendenden Ports:

	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	

So dann das Ergebnis (mit Kext):



PXSX ist meine Fresco Karte, die hatte ich eingebaut, um evtl. das BT Modul darüber laufen zu lassen, das hat aber auch nicht funktionier. Auch ein externen USB mit Adapter zu nutzen hat leider seinerzeit nicht funktioniert.

Meine SSDT-RHUB-USB anbei, diese hat aber nicht funktioniert. USB 3 wurde damit hart abgeschnitten und war nicht im IOReg vorhanden.

Trotz Hilfe der Spezialisten hier konnte keine Lösung gefunden werden.

Der alte Thread HIER

Vielleicht hast Du als "Erdenker" der Idee einen Vorschlag. Alle anderen Daten habe ich per PN gesendet.

Vielen Dank schon einmal.

Viele Grüße

g.com

Beitrag von „Max.1974“ vom 20. November 2024, 02:57

Hallo, wäre es möglich, einen Fenvi PCIe zu verwenden, um natives Bluetooth zu erhalten? Ich verwende es auf meinem Z790 Aorus Elite AX und es funktioniert wunderbar. In meinem SSDT RHUB hat es eine andere Konfiguration als bei Ihnen. Mir ist aufgefallen, dass bei Ihnen die GUPC-Methode nicht vorhanden ist oder nicht angezeigt wird. Ich würde gerne Ihr Original sehen, wenn ich irgendwie helfen kann. Und was das Mapping angeht, habe ich im Jahr 2022 viele Monate damit verbracht, zu lesen und zu lernen, dass es das Problem nicht lösen kann, wenn man den GUPC nicht ändert und 0x03 hinzufügt.

Und ich verwende auch Usb Map zum Zuordnen, aber nachdem es hochgefahren ist, wechsle ich zur Extraktion mit Hackintool und verwende Usbports und SSDTS.

Ich habe den Schlaf mit anderen Methoden gelöst. Ganz anders. Mit SSDTs und Argumenten in der Plist.

Beitrag von „apfelnico“ vom 20. November 2024, 11:41

> Zitat von G.com
>
> Meine SSDT-RHUB-USB anbei, diese hat aber nicht funktioniert. USB 3 wurde damit hart abgeschnitten und war nicht im IOReg vorhanden.

Kann so auch nicht funktionieren, da die Deklaration innerhalb der ACPI längst passiert ist, und

zwar über die "xh_rps14", die auch bei dir enthalten ist. Eine zusätzliche SSDT, die die gleichen Methoden anwendet, wird nicht geladen, denn es dürfen je Device und späteren Scope nur Methoden einmalig vorkommen. Was ja auch logisch ist.

> Zitat von max.1974
>
> Mir ist aufgefallen, dass bei Ihnen die GUPC-Methode nicht vorhanden ist

Ist da, in eben dieser vorgenannten SSDT.

Und die ist auch der Schlüssel. Einfach öffnen und die Table-ID oder Length notieren, diese per OpenCore ACPI dann ausklamüsern.

Dann diese SSDT als Grundlage für eigene Modifikationen nehmen und wieder per OpenCore in ACPI einfügen.

Ich würde es dort einfacher halten, das Konstrukt mit GPLD und GUPC entfernen und jeden Port direkt mit _UPC und _PLD beschreiben, ebenso interne Hubs hinzufügen.

Beitrag von „apfelnico“ vom 20. November 2024, 13:30

kleiner Nachtrag, im verlinkten PDF ab Seite 524 wird _UPC und _PLD sehr gut beschrieben mit Beispielen, auch verschachtelten Hubs. Einzelne, nicht benötigte oder vorhandene Ports werden nicht – wie in einer hier schon verlinkten SSDT per _STA Methode rausgekloppt – sondern innerhalb der _UPC definiert.

ACPI_6.0.pdf

Beitrag von „apfelnico“ vom 20. November 2024, 16:11

G.com

Mit deinen Bildern und deren Beschriftung der Ports kommt einiges nicht hin. Wenn auf der Rückseite die vier USB2 (interner Hub) HS11 sind, können nicht auf dem Deckel des Gehäuses zwei USB2 Ports ebenfalls HS11 sein. Auch scheinen mir die "versetzten" HS/SS Ports der USB3 zumindest nicht sinnvoll.

Ich mach mal einen Vorschlag. Ich habe die relevante SSDT mal bereinigt, für HS11 und HS12 die internen Ports hinzugefügt und nach ACPI-Vorgabe beschrieben. Ansonsten habe ich erstmal ALLE HSxx (Highspeed, USB2) als solche definiert, ALLE SSxx (Superspeed, USB3) als solche definiert.

Damit liegt die Anzahl der Ports natürlich deutlich über das macOS Port-Limit, aber die Definition ist schonmal macOS-leserlich gemacht. Was ich noch nicht weiß, wo die USB-C liegen und konkret welche Variante. Hast du für dein macOS einen gültigen "PortLimitPatch"? Dann nutze den mal, dann werden wir bald schlauer.

Was wäre also zunächst zu tun?

• alle SSDT die du zusätzlich für USB eingebunden hast, müssen raus

• Kexte für USB müssen raus

• die in der ACPI vorhandene SSDT-11 – "xh_rps14" – muss per OpenCore deaktiviert werden

• die beigefügte SSDT muss per OpenCore eingefügt werden.

• macOS Port-Limit aushebeln

Nach Neustart sollten nun sämtliche Ports aktiv sein. Nun bitte noch die restlichen Ports exakt lokalisieren. USB2-Ports sind soweit klar (HS11/12 vervielfältigt durch Hubs), Aber die USB3 inkl. deren USB2 sind interessant. Bei USB-C bitte unbedingt USB3 und USB2 gerät anstecken und jeweils auch den Stecker drehen! Mitunter hat eine USB-C Buchse drei(!) Ports (einen gemeinsamen HS, zwei SS).

Wenn das alles klar ist und immer noch das Portlimit übersteigt, dann eben überlegen, welche Ports ge0pfert werden sollen. Das können wir dann in der SSDT aktualisieren und dann wird auch in der Folge keine USB-Kext benötigt, alle Angaben sind ja schon in der ACPI.

EDIT: Sollte es keine Möglichkeit geben das macOS Port-Limit temporär zu sprengen, dann kann ich die SSDT anpassen, und erstmal (neben HS11/12) die ersten sechs HS/SS aktivieren. Dann liegt mal im Port Limit und kann schauen, welche von den derzeit aktiven Ports auch tatsächlich genutzt werden und korrekt deklarieren, ungenutzte schon entfernen und dann in einem weiteren Anlauf wieder nun freie Ports aufnehmen.

Beitrag von „G.com“ vom 21. November 2024, 19:12

apfelnico

Oh klasse, danke danke!

Ja, hast recht, Front USB 2 hängt ja an dem internen Hub HS12. Ist ein Fehler.

Fakt ist und das ist mir neu, bei dem Board ist eben nicht HS0x gleich SS0x die Endnummern differieren. Das kann ich sicher bestätigen. Hatte unter Windows gleich ohne Opencore und später immer dieselben Abweichungen. Werde das aber erneut sicher stellen und mal unter Windows ohne OC Boot im Hintergrund das USBMap Tool nutzen.

Die USB-C hatte ich seinerzeit mittels Low und High Speed Kabel(USB2/USB3) und meiner Pico 4 identifiziert.!

Ich meine das Limit hatte ich durch XHCI Port Limit Patch und SSDT-USB-Reset realisiert, so ich recht entsinne. Werde ich testen.

VG und nochmal Danke

Beitrag von „apfelnico“ vom 22. November 2024, 08:38

> Zitat von G.com
>
> Fakt ist und das ist mir neu, bei dem Board ist eben nicht HS0x gleich SS0x die Endnummern differieren.

Ja, hatte ich auch noch nicht so gesehen, aber wird genau so sein. Wenn es mit der Zählung schon mit einem USB-C losgeht und dieser drei Ports hat (zwei SSxx, einen HSxx, Connector Type 0x0A (USB2 und SS ohne Switch)), dann verschiebt sich das Ganze schon für die weiteren Ports.

> Zitat von G.com
>
> Ich meine das Limit hatte ich durch XHCI Port Limit Patch und SSDT-USB-Reset realisiert, so ich recht entsinne. Werde ich testen.

Die "SSDT-USB-Reset" lass bitte ganz raus, die haut ja den ganzen Hub des XHCI weg – und somit auch die weiterführenden SSDT, auch die modifizierte.

Ich werde die SSDT anhand deiner Screenshots anpassen. Dann muss auch nix mit dem Port Limit Patch gemacht werden.

Beitrag von „G.com“ vom 23. November 2024, 01:02

Moin apfelnico

ich habe jetzt noch einmal alle Ports unter Windows identifiziert und ganz genau abgeglichen. Meine obige Tabelle ist korrekt.

Auf den Screenshots ist tatsächlich nur der HS11 an der Front korrekterweise der HS12.

Das Problem mit dem Shift entsteht durch den Umstand, dass HS02 zwar definiert aber nicht über die Ports erreichbar sind. Dieser muss intern hart verdrahtet sein

Wo der nun dazugehört entzieht sich meinem Verständnis. Somit wird dann aber ab SS02 einfach bei den HS0x geshiftet.

Vielleicht siehst Du mehr anhand der SSDT, ich bin da noch zu grün.

Das der USB-C drei Ports hat - das kann ich nicht bestätigen, gerne kannst Du mir das noch einmal erklären. Wie kommst Du darauf?

VG

Beitrag von „apfelnico“ vom 23. November 2024, 09:52

USB-C können einen SS und einen HS haben, der SS läuft dann über einen Switch. Damit der Stecker, egal wie rum eingesteckt, auch funktioniert. Dann werden diese Ports als "0x09" definiert.

Es geht auch ohne Switch, dann werden zwei SS je Drehrichtung verdrahtet. Dann bekommen die drei Ports, zwei SS und ein HS eben die Bezeichnung "0x0A".

Vergleiche Portzuweisung aus dem oben verlinkten PDF:

Element	Object Type	Description
Type	**Integer** (BYTE)	Specifies the host connector type. It is ignored by OSPM if the port is not user visible: 0x00: Type 'A' connector 0x01: Mini-AB connector 0x02: ExpressCard 0x03: USB 3 Standard-A connector 0x04: USB 3 Standard-B connector 0x05: USB 3 Micro-B connector 0x06: USB 3 Micro-AB connector 0x07: USB 3 Power-B connector 0x08: Type C connector - USB2-only 0x09: Type C connector - USB2 and SS with Switch 0x0A: Type C connector - USB2 and SS without Switch 0x0B– 0xFE: *Reserved* 0xFF: Proprietary connector
Reserved0	Integer	This value is reserved for future use and must be zero.
Reserved1	Integer	This value is reserved for future use and must be zero.

EDIT:

Deshalb ist bei USB-C eben auch beim USB3-Anteil per um 180º gedrehtem Stecker zu prüfen, ob hierbei ein anderer Port benutzt wird. Schrieb ich glaube ich schon weiter oben.

Beitrag von „G.com“ vom 23. November 2024, 10:26

Moin apfelnico

USB-C wurde entsprechend geprüft. Stecker rau und 180° gedreht und Kabel noch dazu in beiden Richtungen. Bleibt

immer beimselben SS Port. Hat also einen Switch.

VG

Beitrag von „apfelnico“ vom 23. November 2024, 10:33

Ah, OK. Dann gibt es aber noch einen USB-C auf dem Board zum Anstecken an ein Gehäuse. Ein Solcher Port könnte ohne Switch sein, ist bei mir jedenfalls so. Kannste auch noch mal testen.

Beitrag von „G.com“ vom 23. November 2024, 12:47

Anhand der Screenshots kann man sehen, vorne und hinten sin je 1 USBC. Jeweils mit Switch. Der HS02 taucht nirgendwo auf. Egal was ich wo anstecke. Klingt seltsam ist aber so