

Erledigt (OC-config) Lenovo Thinkpad T450 Bluetooth und Soundprobleme

Beitrag von „kuckkuck“ vom 20. Dezember 2019, 19:05

Ich muss mal ein paar Sachen, u.a Missverständnisse richtigstellen. Also eins nach dem anderen:

[Zitat von MacPeet](#)

Hast Du nun Audio mit AppleALC? Mit welcher Id und welchen Ergebnissen/Geräten?

Ja, er hat jetzt (wenn auch rauschendes) Audio mit AppleALC. Wie gesagt, wir haben da nur ganz kurz dran arbeiten können, aber ich habe dir mal den letzten IOReg angehängt, den ich von Obst-Terminator erhalten habe, sowie die "proof-of-concept" DSDT, die ich ihm bereitgestellt hatte.

[Zitat von MacPeet](#)

Ferner wurde hier gesagt, dass die DSDT nicht komplett gepatcht ist und Du die nicht benutzen solltest.

Habe auch nie behauptet, dass ich diese komplett gepatcht hätte.

Erstmal bezog sich das nicht auf deine, sondern auf meine DSDT. Ich habe nicht deine DSDT überarbeitet, sondern mir die Arbeit gespart nachzuvollziehen, was du gemacht hast, und eine neue gepatcht. Diese DSDT ist die "proof-of-concept" DSDT von der ich rede (im Anhang).

[Zitat von MacPeet](#)

Habe nur HDEF eingefügt und einige IRQ's entfernt (IRQ-Patch).

Dann haben wir also praktisch das Gleiche gemacht. Ich habe ebenfalls nur verschiedene _DSM Properties zu HDEF hinzugefügt und verschiedene IRQs überarbeitet. Ich habe [diesen Beitrag](#) verlinkt bekommen und gesehen, dass unter OC die alc-layout-id wohl nicht auf IO Ebene erscheint und zudem mitbekommen, dass verschiedene Versuche die ID zu injecten scheitern. Daraufhin habe ich mich daran gemacht analog zum anscheinend funktionierenden Clover Ordner+IOReg die ID 28 zu injecten. Um zu sehen, ob es überhaupt funktioniert direkt mehrfach (wie gesagt wir hatten keine Zeit) nach folgendem Schema: https://github.com/acidanthera...pplerALC/kern_alc.cpp#L173

Sprich inject von layout-id (wie eh und je, wird zu alc-layout-id gecastet) und alc-layout-id

über `_DSM`, sowie global override durch "alcid" bootarg. Schon hier sieht man, dass die DSDT nur proof-of-concept ist.

[Zitat von MacPeet](#)

Wenn man keine gepatchte einträgt, dann wird immer die Clean geladen, welche auch nicht gepatcht ist.

Ich verstehe nicht genau was du damit sagen willst. Klar wird die in die FW gegossene (dynamische) DSDT geladen, wenn kein Override per Bootloader erfolgt, aber die Idee ist ja die dynamische Struktur der Main-System-Table (DSDT) beizubehalten und diese durch SSDTs zu erweitern...

[Zitat von MacPeet](#)

Diese von mir ist quasi die Clean mit den kleinen Änderungen.

Die von mir ebenfalls, aber hier liegt ja bereits das Problem. Die von uns bereitgestellten DSDTs sind marginal, nicht "komplett" gepatcht und somit nicht als "gepatchtes ACPI" zu bezeichnen. Das heißt es bedarf entweder noch einigen Erweiterungen der DSDT damit auf dem Laptop alles läuft, oder die ACPI Struktur muss noch anderweitig erweitert werden, zB mit SSDTs und Hotpatches. Besonders Renames werden hier problematisch, da OC Hotpatch-Renames vor dem Laden der hinterlegten Tabellen ausführt "Perform binary patches in ACPI tables before table addition or removal.". Vor diesem Hintergrund & wegen den doppelten Injections in meiner HDEF Methode & aufgrund meiner allgemeinen Ablehnung gegenüber gepatchten DSDTs (siehe unten) habe ich dazu geraten meine halb gepatchte proof-of-concept DSDT nicht zu benutzen und stattdessen eine sinnvolle ACPI Struktur aufzusetzen.

[Zitat von MacPeet](#)

Auch nach [Biosupdate](#) lässt sich auf Grund der bereits gemachten Erfahrungen schnell wieder eine DSDT patchen und warum überhaupt ein [Biosupdate](#), sofern alles super läuft.

Eine DSDT ist was Device-States und besonders OperationRegions und die darin enthaltenen MMIO Adressen angeht weitaus dynamischer als das, nicht nur Biosupdates, sondern besonders BIOS-Settings und Boot-environments (OS), sowie Hardware-Veränderungen und BIOS-Patches haben Einfluss auf die DSDT und verändern dieses dynamische Konstrukt. In dem Moment wo wir eine DSDT (über zB MaciASL) auslesen sehen wir den aktuellen Machine-State, das ist ein Snapshot aber nicht mehr und im Falle MaciASL sogar bereits durch das Boot-environment modifiziert. Wenn wir diesen Snapshot als Basis für eine DSDT nutzen, die allgemeingültig sein soll und unter OC selbst mit Windows kompatibel sein muss, sind wir meiner Meinung nach auf dem Holzweg. Der beste Weg sollte deshalb sein die bestehende ACPI Struktur höchstens zu ergänzen und diese Ergänzungen möglichst dynamisch zu

gestalten, und das passiert am allerbesten indem man ACPI Patches umgeht und IO-Patches (siehe WEG & AppleALC) benutzt und wenn es doch ACPI sein muss, dann auf SSDTs zurückgreift und diese mindestens Boot-environment-abhängig gestaltet (_OSI-abhängige Konstrukte um zB Windows-Kompatibilität zu sichern). Und ob das jetzt 20 oder 2 SSDTs sind ist erstmal Präferenz des Users (Übersicht etc.) und der Injection-Zeit-Unterschied von 20 SSDTs gegenüber von 2 sollte erstens marginal sein und zweitens zu vernachlässigen sein gegenüber der Zeit die es braucht den Inhalt der SSDT-Tabellen zu Parsen ganz zu schweigen vom Blocken und Parsen einer kompletten DSDT. Und was Renames angeht, sollten globale Hotpatch-Renames benutzt werden, denn beim Durchführen von Renames in einer DSDT, müssten ohne Hotpatch alle anderen Tabellen, die auch die umbenannten Devices enthalten, ebenfalls gepatcht und explizit gepatcht injected werden, da es sonst zu Device-Name Inkonsistenzen kommen würde, aber ich glaube das ist klar.

Ich hoffe es fühlt sich niemand angegriffen, ich wollte nur meine Gedanken zu dem Thema loswerden und wenn es andere Erkenntnisse gibt lasse ich mich auch gerne eines besseren belehren 😊 Ansonsten angenehme Weihnachtsferien an alle!