

RX 6600 auf bestehendes Z490 UHD630 Sonoma - Frage: UHD Passiv oder OFF - SMBIOS Empfehlung - best Practice ?

Beitrag von „byebye123“ vom 27. November 2023, 17:19

[Zitat von Nordel](#)

Richtig. Die Karte muss aktiviert sein. Dass sie Headless läuft, legst du über die config fest.

gene-x Wenn ich nicht irre: Wenn du MacPro7,1 wählst, solltest du noch CPUfriend.kext (o.s.ä) berücksichtigen. Der MacPro7,1 kennt ja von Haus aus deine CPU nicht.

EDIT:

unterm strich lässt sich sagen das die Nutzung des CPUfriend.kext etwas problematisch ist da hier mit jedem Update Probleme entstehen könnten und auch wahrscheinlich sind.

Lässt sich wohl durch deaktivieren des kext vor Update und neu kompilieren des Sub Kext regeln aber ist wohl so da hier auf Daten Basis des Systems etwas erzeugt wird das Änderungen unterworfen ist.

Die Piker-Alpha Methode funktioniert nicht immer perfekt aber wenn dann dann dauerhaft.

Mal sehen ob da die Passiv UHD mit SMBIOS iMac nicht solider ist.

Hinzu kommt das IntelPowerGadget nicht mehr weiterentwickelt wird und jetzt Terminal oder Grafana Anbindung von PCM benötigt wird.....how complicado....

Stimmt, da war doch was 😊 Das kenne ich noch vom Laptop mit PikerAlpha da musste ich das manuell eintragen da es nicht in der Datenbank war I7-8550.

Das folgende finde ich nicht wirklich erhellend weil ich nicht weiss welche CPU Daten für den

CPUFriendDataProvider.kext denn zur Verfügung gestellt werden.

NOTE: For simplicity, `CPUFriendDataProvider.kext` is always preferred. Where there is another SSDT generated by [ssdtPRGen.sh](#), combination between `ssdt_data.dsl` and the SSDT table produced by this script is required. See [Data Combination](#) for further details.

Auf dem Laptop habe ich nur die `ssdt.aml` mit Piker-Alpha generiert genutzt aber das ist schon was her.....

Im Prinzip sollte das doch immer noch so gehen:

[Intel Speedstep CPU](#)

Code

1. Bisher war es ja meist so das es hin zur `ssdt.aml` ging und damit kam man um `ssdtPRGen.sh` nicht herum.
- 2.
3. Warum auch es macht halt das was es soll.
- 4.
5. Das geht mit der Beta (enthält aktuelle CPU's) auf folgende Art
- 6.
7. Code
8. `curl -o ~/ssdtPRGen.sh https://raw.githubusercontent.com/Piker-Alpha/ssdtPRGen.sh/Beta/ssdtPRGen.sh`
- 9.
10. Code
11. `wc -c ssdtPRGen.sh`
- 12.
13. Rechte der Datei setzen
14. Code
15. `chmod +x ~/ssdtPRGen.sh`
- 16.
17. Hilfe der Befehle einsehen
18. Code
19. `~/ssdtPRGen.sh -h`
- 20.
21. Script starten (ist eigentlich selbst erklärend)
22. Code

23. ~/ssdtPRGen.sh

Alles anzeigen

Code

1. --kext "file" Erstellt den CPUFriendProvider.kext mit Informationen, die von "file" bereitgestellt werden.
2. --acpi "file" Erstellt die ssdt_data.dsl mit Informationen, die von "file" bereitgestellt werden.

Das macht dann schon eher sinn, super Doku von CPUfriend, welches "file" ?

Die Variable "file" ist die plist von
/System/Library/Extensions/IOPlatformPluginFamily.kext/Contents/PlugIns/X86PlatformPlugin.kext/Contents/R

Code

1. cd ~/Desktop

Code

1. git clone <https://github.com/PMheart/CPUFriend>

Code

1. cd ./CPUFriend/ResourceConverter

Code

1. ./ResourceConverter.sh --kext
/System/Library/Extensions/IOPlatformPluginFamily.kext/Contents/PlugIns/X86PlatformPlugin.kext/Cont
9AE82516C7C6B903.plist

Code

1. ls
2. CPUFriendDataProvider.kext ResourceConverter.sh

Die hier erstellte Kext stammt vom SMBios MacBook9,1 Mac-9AE82516C7C6B903

Alle vorher benutzen ssdt usw. sollten dabei entfernt werden.

[Zitat von @griven](#)

Es ist einfach eine andere Herangehensweise als die die PikerAlpha mit dem SSDTPRGen wählt. Das SSDTPRGen liest den Prozessortypen aus und erzeugt anhand der von Intel zu dem Prozessor bereitgestellten Daten eine SSDT die alle Steps enthält die von der CPU unterstützt werden (Min und Max und Max Turbo) Hierbei werden sowohl die C als auch die P States erzeugt und in der SSDT verewigt. Eine mit dem SSDTPRGen Script erzeugte SSDT allein ist noch kein Garant dafür das der Speedstep unter OS-X auch richtig funktioniert hier bedarf es noch einer zweiten Komponente nämlich eines passenden PlattformPlugins. Welches Plugin macOS hier wählt hängt vom eingesetzten SMBIOS ab. Abhängig vom gewählten SMBIOS wählt macOS aus den Ressourcen der X86PlattformPlugin.kext das zum Modell passende Profil aus und steuert anhand dieser Vorgaben sowohl das CPUPowerManagement als auch die Fähigkeiten die ein MAC im Sleep besitzt (Darkwake, Povernap usw.). Geht man bei der Auswahl des SMBIOS mit Bedacht vor kommt man auf die Weise leicht zu einer Konfiguration die zufriedenstellend funktioniert.

Der CPUFriend dreht an der Stelle den Spieß einfach um und erzeugt eine SSDT oder eben einen HelperKext nicht aus den zur eingebauten CPU passenden Informationen sondern aus den Vektoren die im PlattformPlugin definiert sind sprich das Ergebnis des CPUFriend ist genauer auf das jeweils verwendete SMBIOS abgestimmt und definiert dann vermutlich auch nur die wirklich nötigen Steps. Der Weg ist ein anderer aber das Ergebnis unter Strich wohl das gleiche 😁

SMBIOS iMac 20,1:

Mac-CFF7D910A743CAAF

Find the correct SMBIOS File for the Script in:

`/System/Library/Extensions/IOPlatformPluginFamily.kext/Contents/PlugIns/X86PlatformPlugin.kext/Contents/R`

Woher bekomme ich die passende Bezeichnung für die diversen Files passend zum SMBIOS im Ordner:

<https://dortania.github.io/Ope.../smbios-support.html#imac>