

10th Gen Intel i5/i7/i9 CPU Power Management optimieren

Für 10th Gen Intel CPUs ist das iMac20,2 SMBIOS am besten geeignet, da diese Modelle mit i7/i9 CPUs ausgestattet sind. Falls ein i5 verwendet wird, bitte iMac20,1 als SystemProductName verwenden.

Da ich einen i9 habe, verwende ich iMac20,2. Das System läuft damit flüssiger als mit iMac19,1. Die Leistung ist ebenfalls besser und die idle-Frequenz ist ebenfalls niedriger. Sie kann aber mit Hilfe von CPUFriendFriend noch weiter gesenkt werden.

Allerdings muss diese Optimierung auf der verwendeten Maschine durchgeführt werden - es sei denn, es wird eine i9 10850K CPU verwendet. In diesem Fall müssen lediglich CPUFriend und CPUFriendDataProvider.kext in der config.plist aktiviert werden. Ansonsten müssen Sie eine eigene CPUFriendDataProvider.kext generieren, die für Ihre CPU geeignet ist.

Dazu überprüfen wir zunächst die aktuelle Board-ID, indem wir den folgenden Befehl im Terminal ausführen: **ioreg -l | grep -i board-id**

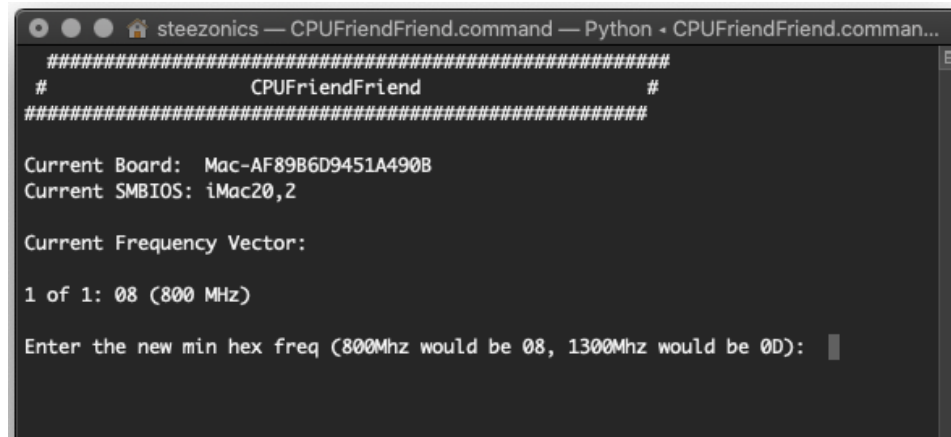
Ergebnis für iMac20,2 sollte sein:
| "board-id" = <"Mac-AF89B6D9451A490B">

Ergebnis Für iMac20,1 sollte sein:
| "board-id" = <"MMac-CFF7D910A743CAAF">

Falls das Ergebnis abweicht, prüfen Sie bitte, ob unter PlatformInfo > Generic > SystemProductName iMac20,1 oder iMac20,2 eingetragen ist, korrigieren es entsprechend der verwendeten CPU, speichern Sie die config.plist und starten neu, um die Änderungen des SMBIOS anzuwenden.

CPUFriendFriend verwenden

Als nächstes laden wir CPUFriendFriend herunter (Code > Download ZIP) und führen *CPUFriendFriend.command* aus. Folgendes sollte zu sehen sein:



```
steezonics — CPUFriendFriend.command — Python · CPUFriendFriend.comman...
#####
#                               #
#####

Current Board:  Mac-AF89B6D9451A490B
Current SMBIOS:  iMac20,2

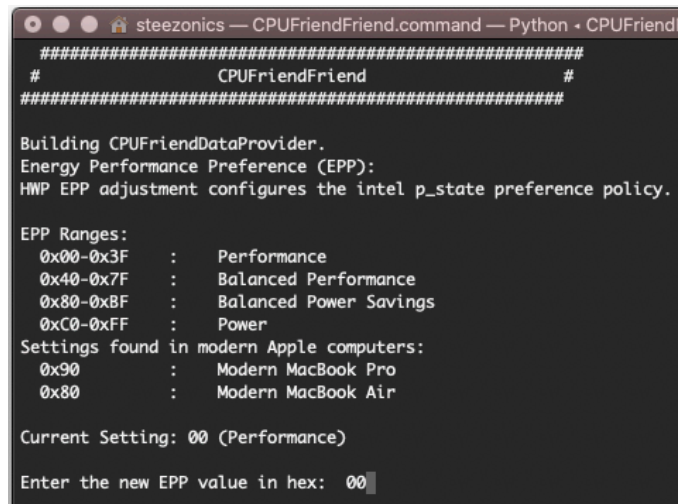
Current Frequency Vector:

1 of 1: 08 (800 MHz)

Enter the new min hex freq (800Mhz would be 08, 1300Mhz would be 0D):
```

Wir tragen **08** ein für **800 mHz** Leerlauf-Frequenz (idle).

Im nächsten Fenster geben wir die **Energy Performance Preference (EPP)** als Hexadezimalwert an. Diese beschreibt, wie schnell von der niedrigsten zur höchsten Turbo-Frequenz skaliert wird, was Auswirkungen auf den Stromverbrauch hat. Es gibt hier 4 Bereiche: **0x00**, **0x40**, **0x80** und **0xC0**. Stellt den Wert entsprechend eurer Bedürfnisse ein.



```
steezonics — CPUFriendFriend.command — Python · CPUFriendFriend
#####
#                               #
#####

Building CPUFriendDataProvider.
Energy Performance Preference (EPP):
HWP EPP adjustment configures the intel p_state preference policy.

EPP Ranges:
0x00-0x3F : Performance
0x40-0x7F : Balanced Performance
0x80-0xBF : Balanced Power Savings
0xC0-0xFF : Power

Settings found in modern Apple computers:
0x90 : Modern MacBook Pro
0x80 : Modern MacBook Air

Current Setting: 00 (Performance)

Enter the new EPP value in hex: 00
```

Als nächstes muss man die **Performance Bias** einstellen, mit der man die generelle Ausrichtung des Systems zwischen Leistung und Energieeffizienz vorgibt.

Die Skala reicht von **00** (maximale Performance) bis **15** (maximaler Energiesparmodus). Da dies eher für Notebooks als für eine i9 Workstation von Interesse ist, habe ich den Wert auf 0 gesetzt.

Nachdem man die gewünschte Ausrichtung des Systems angegeben hat, werden Frequenzvektoren aus der zur Board-ID zugehörigen .plist ausgelesen, angepasst und mehrere Dateien erzeugt. Darunter sowohl eine »ssdt_data.aml« als auch die Kernel Extension »CPUFriendDataProvider.kext«.

Diesen Kext kopieren wir zusammen mit CPUFriend.kext in den Kext Ordner von OpenCore, erstellen mit Hilfe von ProperTree einen neuen Snapshot der Config und starten den Rechner neu.

Danach IntelPowerGadget starten und den Frequenzverlauf beobachten. Sowohl **CoreMax** (oberste Linie) als auch **CoreAvg** (zackige blaue Linie) sollten sich im Leerlauf tendenziell nach unten bewegen, wobei die obere Linie nur maximal bis zur durchgezogenen Geraden (= Base Frequency) fallen kann, was bedeutet, dass die CPU gerade nicht boostet. Am unteren Ende der Skala, sollte **CoreMin** unter 1 GHz liegen, da wir ja 800 MHz als LFM-Wert eingetragen haben. In meinem Fall liegt sie sogar noch 100 mHz darunter bei 0,7. Somit funktioniert das CPU Power Management.

Falls die Frequenz nie unterhalb der Base Frequency oder immer darüber liegt, ist dies ein Indiz dafür, dass etwas mit dem Power Management nicht stimmt.

