

Anleitung: NVMe SSD per SSDT als Boot-Laufwerk

Beitrag von „maschinenwart“ vom 6. März 2017, 00:15

Vorwort:

Die Einbindung der SSD auf diesem Wege ist zwar etwas aufwändiger, dafür aber sehr erfolgreich. Die meisten Informationen hierfür habe ich von „[RehabMan](#)“, „[Pike R. Alpha](#)“ und „[Mork vom Ork](#)“.

Auf der github.com-Projektseite von RehabMan gibt es dann einen Link auf TomatenTony's Forum, in dem RehabMan das sehr ausführlich erklärt. Allerdings alles in englisch und teilweise sehr kompliziert. Deshalb mein Versuch, dieses leichter verständlich auf deutsch zu erklären

...die Tomate ist übrigens ein Apfel... 

Hier im Forum gibt es schon eine ähnliche [Anleitung von lupotmac](#), da wird die Methode mit KextToPatch-Einträgen beschrieben bzw. eine Methode bei der man die originale IONVMeFamily.kext löschen muss.

Bei dieser Anleitung hier kann man die originale IONVMeFamily.kext unbeachtet lassen und die Patch-Einträge fallen auch weg!

macOS Version: 10.12.3

Zuletzt bearbeitet: 22.03.2017

Was wird gebraucht:

Das erkläre ich an Hand meines Systems. Es gibt viele vergleichbare Komponenten, mit denen

das wahrscheinlich genauso funktioniert. Wichtig ist es, keine zu exotischen Bauteile zu verwenden. Bei der Auswahl meiner Komponenten, wie Chipsatz, CPU, SSD und Grafik habe ich mich an diverse Empfehlungen aus dem Netz gehalten.

Spoiler anzeigen

Los geht's:

Hinweis: zur Bearbeitung von .plist-Dateien bitte keinen Text-Editor benutzen. PlistEdit Pro oder Xcode sind gefahrloser und machen die Dateien nicht unbrauchbar.

Hinweis: habe festgestellt, dass ein Reboot nach Änderungen mit dem Clover Configurator oft nicht ausreicht. Da hilft nur, Rechner runterfahren (ausschalten) und wieder neu starten.

Hinweis: falls du an einem bestehenden System arbeitest - Backup Sicherung nicht vergessen!

Die grünen Menüpunkte sind erforderlich, die schwarzen Menüpunkte sind optional:

1. ggfs. Bootstick für Windows erstellen

- falls nur eine ISO aber keine Installations CD vorliegt

2. BIOS update

- Auf der Hersteller-Seite das neueste BIOS (F20) runtergeladen und geflasht.

3. Installiere Windows

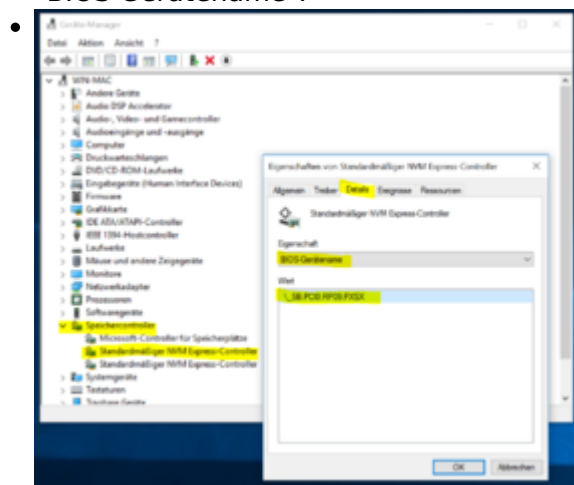
- Um die SSDT erstellen zu können, musste ich zuerst Windows auf dem Rechner installieren und zusätzlich alle Treiber für das Board, damit alle Controller und Geräte im Gerätemanager auftauchen. Das habe ich auf einer der zwei SSD gemacht.

4. ggfs. Firmware-Updates durchführen

- Um meinen Thunderbolt-Controller nutzen zu können musste ich dafür ein Firmwareupdate durchführen. Das ging nur unter Windows! Deshalb habe ich auch alle anderen Treiber installiert und jedes Gerät unter Windows überprüft.

5. Finde den ACPI Pfad der NVMe SSD

- In - Systemsteuerung > Geräte manager > Speicher Controller > Standard NVM Express Controller > Eigenschaften > Details - findet man den ACPI Pfad. Hierfür sucht man den "BIOS Gerätenamen".



- Der Pfad sieht bei mir so aus:

_SB.PCI0.RP09.PXSX

- Dieser ist von Board zu Board und dem jeweiligen Steckplatz, falls mehrere vorhanden sind, unterschiedlich, deshalb wirklich selber suchen.

6. Bootstick erstellen (macOS)

- Einen USB-Installationsstick habe ich mir zuvor bei einem Freund am Mac erstellt. Anleitungen dafür gibt es auch [hier im Forum](#).

7. Installiere macOS Sierra

- Um die SSDT zu erstellen brauchen wir einen Rechner mit macOS also habe ich das

schnell mal installiert. Ich habe ja noch eine zweite SSD zur Verfügung. Die Windows SSD abgestöpselt und die leere SSD angestöpselt. Eine Installationsanleitung gibt es [hier im Forum](#) oder mal Google fragen.

8. Erstelle die SSDT-NVMe-Pcc.aml, hierfür habe ich [MaciASL](#) verwendet.

Spoiler anzeigen

- Eine Vorlage im DSL-Format findest du im Anhang dieser Anleitung.
- Vor dem Kompilieren überprüfen, ob alles richtig eingestellt ist:
- MaciASL > Preferences > iASL auf ACPI 6.1 setzen
- Die SSDT-NVMe-Pcc.dsl Vorlage in maciASL öffnen
- ACPI **Pfad** anpassen
- Die angepasste SSDT-Vorlage speichern:
- Save As > „SSDT-NVMe-Pcc“
- **Format: ACPI Machine Language Binary**
- Vorzugsweise auf dem Desktop speichern, damit du sie leicht findest

8a. Die fertige Datei wird jetzt in folgenden Ordner der EFI-Partition des USB-Stick kopiert:

- EFI/Clover/ACPI/patched/SSDT-NVMe-Pcc.aml

9. Einen Patcheintrag in die config.plist eintragen - Change all DSM to XDSM - **WICHTIG!**

- Entweder mit dem Clover Configurator unter
 - ACPI
 - DSDT
 - Patches

Comment: Change all _DSM to XD SM

Find: 5f44534d

Replace: 5844534d

- eintragen oder...

Spoiler anzeigen

10. HackrNVMeFamily-*.kext erstellen

- Hierfür habe ich im Terminal folgende Befehle ausgeführt:

Code

1. cd Desktop
2. git clone https://github.com/RehabMan/patch-nvme.git patch-nvme.git
3. cd patch-nvme.git

...hier wird das Paket auf den Desktop geladen.

Code

1. cd ~/Desktop/patch-nvme.git
2. ./patch_nvme.sh --spoof 10_12_3

...hier wird die Kext-Datei generiert und auf dem Desktop abgelegt.

- 10_12_3 ist meine Version von Sierra, die Zahl im Befehl muss an die eigene macOS-Version angepasst werden.
- Es werden aber nur Patch-Skripts für die Versionen macOS 10.11.5 bis 10.12.3 bereitgestellt!

- Das kann dann auch mal einen exotischen Namen haben, z.B. **10_11_6_beta4**
- also unbedingt mit der eigenen Version vergleichen und bei [RehabMan](#) nachschauen!

10a. Generierte "HackrNVMeFamily-*.kext" auf den USB-Installationsstick in folgenden Ordner kopieren:

- EFI/Clover/kexts/other

11. macOS Sierra auf NVMe SSD installieren

- Nachdem ich mit dem vorbereiteten Bootstick den Rechner gebootet habe und im Installationsbildschirm gelandet bin, kann ich im Festplattendienstprogramm meine NVMe SSD auswählen und wie gewohnt die Formatierung bzw. Die Installation durchführen.
- Nach erfolgreicher Installation folgende Dateien vom USB-Stick in den EFI-Ordner der NVMe SSD kopieren:
 - EFI/Clover/ACPI/patched/SSDT-NVMe-Pcc.aml
 - EFI/Clover/kexts/other/HackrNVMeFamily-*.kext
 - EFI/Clover/Config.plist

12. macOS Sierra updaten

- Vor einem macOS-Update sollte man auf das Update vom "patch-nvme github project" warten, um dann die HackrNVMeFamily*.kext mit der neuesten Version zu tauschen.

...diese Anleitung werde ich immer wieder erweitern bzw. anpassen. Falls Fehler vorhanden sind, bitte Bescheid sagen, die werden dann umgehend berichtigt.

Beitrag von „Thogg Niatiz“ vom 13. März 2017, 11:19

Danke für die Anleitung! Ein Frage zu Punkt 10: muss die HackrNVMeFamily.kext an dem entsprechenden Rechner mit der NVMe SSD generiert werden oder kann man die an jedem

beliebigen Rechner erstellen und dann direkt Sierra auf die NVMe SSD installieren ohne die erste Installation auf eine AHCI SSD?

Beitrag von „maschinenwart“ vom 13. März 2017, 17:05

...gerne! 😊

Das kann ich dir nicht mit Gewissheit sagen. Es könnte sein, dass vor der Ausführung des Code abgefragt wird, ob die macOS-Version kompatibel zur gewünschten Kext ist. Ich denke aber, es muss nicht der Installationsrechner sein. Letztendlich werden in der Kext die gleichen Einträge verwendet, die sonst beim CloverConfigurator unter "KextToPatch" eingetragen werden.

mw

P.S.

...ist tatsächlich von der macOS-Version abhängig... <http://www.insanelymac.com/forum/topic/2382465-sierra-is-ready/?p=2382465>

Wenn du 10.12.3 installieren möchtest, könntest du theoretisch meine Kext aus dem Anhang benutzen. Rechnerspezifisch ist aber auf jeden Fall die SSDT.

Habe gestern Nacht eine zweite NVMe SSD eingebunden. Diese ist auf einer PCIe-Karte von ASUS aufgesteckt. Einfach den zweiten Pfad in die SSDT eingetragen, den ich zuvor unter Windows ermittelt habe, kompilieren - fertig!

Beitrag von „filou77“ vom 15. März 2017, 17:57

ich hab mal ne anfänger-frage..

wie müsste ich denn den ACPI pfad korrekt in die vorlage eingeben, wenn er ihn mir mit @

zeichen z.b. über das tool IO Registry Explorer folgendermaßen ausspuckt..?

`_SB/PCI0@0/IOU2@10000/I2PS@0/PPB4@40000/PXS4@ffff`

(angeschlossen ist die NVMe extern über PCI..) ersetze ich die @ zeichen durch punkte geht es auch nicht. und lasse ich die @ zeichen und alles dahinter weg, also so

`_SB/PCI0/IOU2/I2PS/PPB4/PXS4` ,schön kurz wie bei dir, dann will MaciASL es auch nicht annehmen. hab es auch schon mit backslash ´s versucht. hm..

Beitrag von „maschinenwart“ vom 17. März 2017, 01:25

[Zitat von filou77](#)

wie müsste ich denn den ACPI pfad korrekt in die vorlage eingeben

...das weiß ich leider auch nicht.

Beitrag von „filou77“ vom 1. Mai 2017, 20:44

So war es bei mir richtig: `_SB.PCI0.IOU2.I2PS.PPB4.PXS4`

Funktioniert einwandfrei! Richtig tolle Methode. 😊

Ich weiß jetzt auch zumindest annähernd, inwieweit die SSDT abgeändert werden müsste, falls Probleme auftauchen.. Bridge dazwischen etc. 😊

Cool ist.. man lädt quasi die NVMe schon bevor irgendwas anderes geladen wird.

TOP!

Boards dürften allerdings wohl nicht älter sein als 97er Chipsatz habe ich irgendwo gelesen.

Beitrag von „stmeyer“ vom 12. Juni 2017, 20:42

Hi,

ich habe eine NVMe 960 pro und hatte gehofft, dass das howto damit auch funktioniert. Macht es leider nicht.

Liegt das an dem anderen Modell oder mach ich mal wieder was falsch? Das readme von RehabMan habe ich gelesen, die 960er taucht da leider nicht auf...

Gruß

Stefan

Beitrag von „griven“ vom 16. Juni 2017, 22:31

Ich würde das so nicht mehr machen denn inzwischen ist es doch wesentlich einfacher und komfortabler über die Kext2Patch Funktionalität von Clover zu lösen. Warum also erst umständlich eine SSDT bauen und einen zusätzlichen Kext nutzen wenn es quasi mit Bordmitteln genau so gut geht?

Beitrag von „Hecatomb“ vom 15. August 2022, 19:51

Hey hab grad dein thead per Google gefunden, gelesen und gesehen das du sogar online bist.

Verstehe ich das richtig das es hier um ssdt erstellen geht, wenn im macOS installer die

controller bzw. die nvme und ssd nicht da sind?

Jemand hier aus forum hat nämlich dieses problem beim Monterey installer. bei Big Sur sieht er allerdings alle platten.

es geht um diesen user....link zum thread:

[Monterey Installer findet keine Festplatte](#)

Beitrag von „maschinenwart“ vom 16. August 2022, 19:44

...das funktioniert hier leider nur mit "El Capitan" bzw. "Sierra"...