

Komplettaufbau eines El Capitan Hackintosh GA Z97m-D3H

Beitrag von „Hessi“ vom 16. März 2016, 23:02

Hallo Leute,

hier der Aufbau meines Hackintosh (Gigabyte GA Z97m-D3H).
Ist etwa arg umfangreich für das Forum, daher hier nur die reinen Installationschritte.

Die komplette Anleitung findet Ihr auf Tote-Pixel.de





Installation OSX

Boot-Stick erstellen

Auf einem Mac oder Hackintosh den El Capitan-Installer aus dem AppStore laden, nicht installieren!

Mit DiskmakerX einen USB-Boot-Stick (min. 8GB) davon erstellen lassen:

<http://lionsdiskmaker.com>

Das Ding ist absolut selbsterklärend.

OSX installieren:

Vom Stick booten, ggf. mit F12-Taste während des Startvorganges, und El Capitan normal

installieren.

Partition anlegen:

Dienstprogramme --> Festplattendienstprogramm:

Platte auswählen, Partitionieren anklicken

Eine Partition mit einer Größe von 40 bis 100 GB erstellen.

Diese "El Capitan" nennen.

OSX auf "El Capitan" installieren.

Damit wird auch der ganze EFI-Kram installiert.

User anlegen.

Keine Netzwerkverbindung wählen.

Zeitzone später einstellen, wird aktuell nicht vom Installer übernommen

Nach der ersten Anmeldung:

TRIM für die SSD im Terminal aktivieren:

```
sudo trimforce enable
```

SIP umgehen:

Beim folgenden Neustart wieder in das Bootmenue mit F12

"Recovery HD 10.11.3" auswählen

Menuepunkt Dienstprogramme --> Terminal

```
csrutil disable
```

```
reboot
```

Warum das?

Sound und Netzwerk werden nicht OOTB funktionieren. Ebenso werden einige USB

3.0 Ports an der Rechnerrückseite nicht funktionieren.

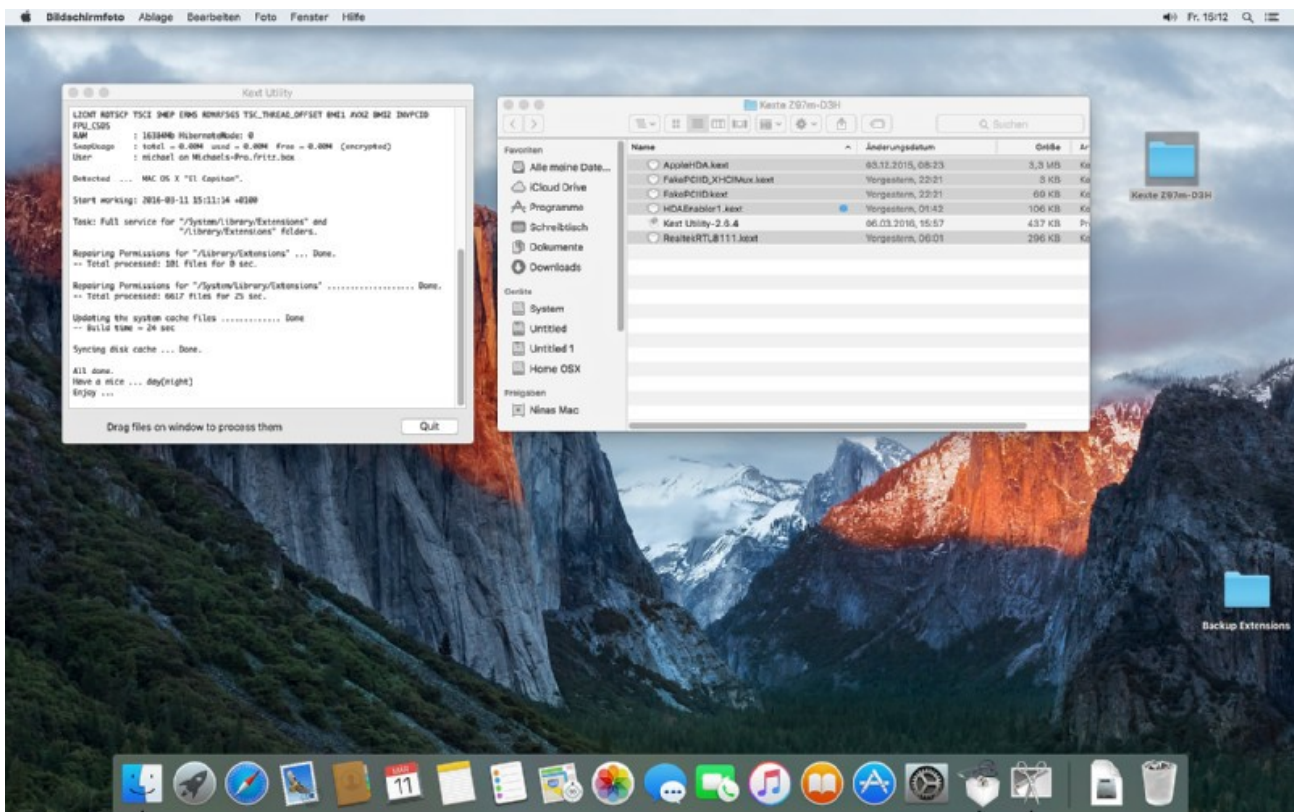
Dafür diese Kexte laden.

Aber [SIP \(System Integrity Protection\)](#) verhindert Änderungen am System. Unsere Kernextensions "Kexte" (quasi Treiber) sind nicht signiert und werden vom System als unsicher eingestuft.

Wir müssen [SIP](#) also erst deaktivieren. Das geht nicht unter OSX selber, sondern nur in der Recovery.

Nach dem dies passiert ist, bootet OSX wieder (ist aber ungeschützt, also nichts aus dubiosen Quellen in der Zeit installieren!).

Kexte installieren:



Mit dem [Kext Utility](#) jetzt [diese Kexte](#) installieren. Das Tool macht das alles automatisch für uns. Sehr praktisch!

Reboot

Optional, wenn wir schon [SIP](#) deaktiviert haben, ist es eine gute Gelegenheit, das Schreiben auf NTFS-Laufwerke zu ermöglichen. OSX kann NTFS von Hause aus nur lesen. Zwar kann man die Fähigkeit des Schreibzugriffs auch im Terminal aktivieren, aber das ist von Apple aus gutem Grund nicht für den User freigegeben. Wir installieren am besten OSX Fuse nach [dieser Anleitung](#). Das ist zudem kostenlos.

Edit: Ich behaupte jetzt einfach mal, dass Fuse bei mir zu Instabilitäten führte. Ich bekam Systemfreezes beim Kopieren eines TM-Sparesbundles vom NAS auf eine externe USB-HDD mit HFS+, auch wenn keine NTFS-Partition beteiligt war. Der Freeze tritt dann auch, wenn Last auf USB und LAN ist und ich noch weiter arbeite. In meinem Fall Bildoptimierung mit DXO Optics Pro. Das System kopiert noch eine Weile weiter und geht dann in den Reboot. Derweil halt der drehende Ball. Tritt auf, wenn man irgendwas anklickt und ein Plattenzugriff ausgelöst wird.

ACHTUNG!

Wieder beim Neustart in die Recovery gehen und im Terminal

```
crsutil enable  
reboot
```

eingeben, damit das System wieder sicher ist!

Fertig! 😊

Die Kür:

Für spätere Erweiterungen, persönliche Anpassungen an den Einstellungen des Systems benötigt man Zugriff auf die EFI-Partition, die man aber normaler Weise im Festplattendienstprogramm nicht sehen und daher auch nicht aktivieren kann. Der wichtigste Grund ist aber: Bei einem Update werden Euch wieder Network, Sound und einige USB-Ports deaktiviert werden. Wir erinnern uns: unsere Kexte sind nicht signiert!

Können wir das SIPUtility überhaupt benutzen? Ein Terminal öffnen und folgenden Befehl eingeben (geht nur mit dem alten FDP von Yosemite, das man sich in den Programmordner kopiert):

```
defaults write com.apple.DiskUtility DUDebugMenuEnabled 1
```

Das FDP wieder starten und im Menü unter "Debug" "Jede Partition einblenden" auswählen. Dann die EFI-Partition aktivieren.

Im Terminal folgenden Befehl eintippen:

```
sudo nvram 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:test=OK  
reboot
```

Nach dem Neustart:

```
nvram 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:test
```

Ausgabe sollte sein:

```
4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:test OK
```

Stimmt die Ausgabe?

Dann ist alles okay und der NVRAM ist beschreibbar.

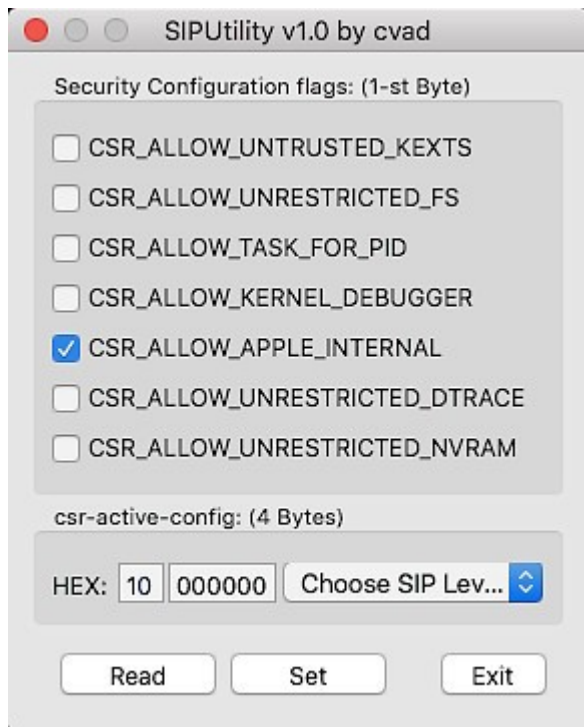
Wir benötigen jetzt noch das [SIP-Utility](#), damit wir nicht jedesmal in die Recovery HD booten müssen, um [SIP](#) zu deaktivieren, wenn ein Update uns die Kexte deaktiviert.

Im

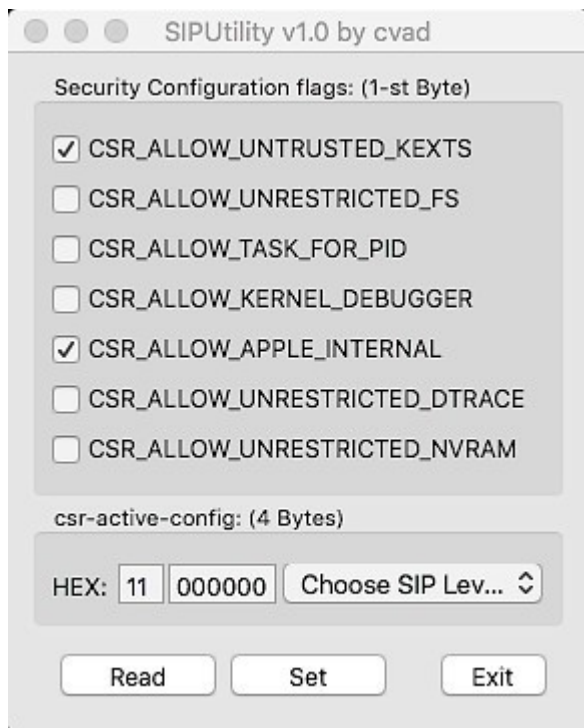
[SIP-Utility](#) können wir die Berechtigungen über eine graphische Oberfläche einstellen. Das ist schon ganz nett. Wir können dort auch einzeln einstellen, dass z.B. nur die Kexte nicht geschützt werden

sollen.

Aber bitte nicht dazu verleiten lassen, diesen Schutz nicht wieder einzuschalten!



(Standardeinstellung)



(Fremd-Kexte erlaubt)

Oberste Feld auswählen, "Set" klicken und neu starten.

Startet das SIPUtility nicht?

Systemeinstellungen --> Sicherheit

Dort

kann man es jetzt einmalig zum Starten erlauben. Oder man setzt die Sicherheitseinstellungen auf "Keine Einschränkungen". Das ist aber nicht unbedingt empfehlenswert.

Fixe für Hardware:

HD4600 Multimonitorbetrieb:

Im Terminal eingeben:

```
sudo nvram 1F8E0C02-58A9-4E34-AE22-2B63745FA101:AAPL,ig-platform-id=220332035
```

Windows und Ubuntu installieren:

Windows:

Zuerst

installieren wir Windows. Im Installer erstellen wir im freien Bereich des SSD eine Partition von 40 bis 100 GB für unser Windows. Je nach eigenen Vorlieben und der Größe der SSD.

Der Rest mit den Windows-Recovery- und dem Windows-Bootmanager-Partitionen macht der Installer automatisch.

Ubuntu:

Jetzt

installieren wir Ubuntu oder Elementary. Im Installer die Partitionierung nicht automatisch erledigen lassen. Selber machen!

Root "/" bekommt 20 GB, das reicht. Ext3-Dateisystem

Swap bekommt 16GB, bzw. die Größe des RAMs des Rechners. Swap-Dateisystem

Home bekommt soviel, wie Ihr benötigt. NTFS-Dateisystem (dann können da alle OS drauf

zugreifen)

Kein Bootloader?

Nein.

Alle OS bringen ihren eigenen Bootloader mit.

Bei Start treffen wir einfach unsere Auswahl über F12, der Bootauswahl.

Diese könnten wir auch im BIOS noch anpassen und in der Reihenfolge ändern.