Anleitung: OpenCore mit Z390 Aorus Pro, i7 8700K , AMD Vega 56

Beitrag von „iBoy“ vom 20. September 2019, 20:37

Hallo Leute,

heute schreibe ich mal eine Anleitung für die OSX Installation mit Open Core für "Anfänger".

Vorab: Ich bin auch neu in der Hackintosh Szene und kann daher nicht sicherstellen, dass meine Tipps alle funktionieren.
Ich kann auch nicht sagen ob alle Einstellungen so optimal sind, aber mein System läuft ohne offensichtliche Probleme.

ACHTUNG: OpenCore ist noch in der Beta und verlangt einiges an Einarbeitungszeit.

Wie immer gilt: Alles auf eigene Gefahr. Ich übernehme keine Haftung.

1.) Meine Komponenten

Mainboard: Gigabyte - Z390 Aorus Pro

CPU: i7 - 8700k

GPU: Gigabyte - AMD Vega 56

RAM: 16 GB

Festplatte: SSD EVO PLUS 500 GB (Achtung diese benötigt ein Firmwareupdate vor der Installation von OSX)

OSX Lizenz

2.) Verwendete Software

TINU (findet ihr im Forum)

OpenCore EFI Maker (1.2.0) (findet ihr im Forum)

Clover Configurator (nur um EFI zu Mounten)

X-CODE

3.) USB Installationsstick erstellen

a) OSX aus dem AppStore laden

b) TINU laden

c) mithilfe von TINU OSX auf den Stick installieren.

Dann hab Ihr einen OSX Installationstick der auf einem Apple PC funktioniert.

Achtung der Stick wird dabei formatiert und alle Daten werden gelöscht.

4.) OpenCore EFI Maker

Damit die Installation auch auf einem PC funktioniert müssen wir einen Bootloader installieren. Hierzu verwenden wir OpenCore, einen noch relativ neuen Bootloader, der zwar noch in der Beta steckt aber dennoch gut läuft.

a) OpenCore EFI Maker herunterladen

b) OpenCore EFI Maker ausführen. Die Einstellungen können wir wie folgt wählen:

1 (OC 0.5.0)

NO (legacy boot)

YES (FileVault)

1 (HfsPlus.efi)

c)Im Anschluss wird ein Ordner auf dem Desktop erstellt.

5.) Config.plist anpassen

a) In dem unter Punkt 4 erstellten Ordner findet Ihr eine Datei namens FullSampe.plist

b) Diese öffnen wir mit X-CODE und passen folgende Parameter an:

ACPI;Add;0;**YES**

Die Patches bekommen wir aus einer standard config aus dem Forum) (ist eigentlich für Clover aber funktioniert auch für OC) (https://www.hackintosh-forum.de/forum/thread/38883-install-stick-für-diverse-system-varianten-high-sierra-10-13-6-und-mojave-10-14/)

Hier kopieren wir alle Einträge aus dem Bereich ACPI ; DSDT; Patches und fügen sie in unsere config unter ACPI; Patch ein.

ACPI;Patch; GFX0 to IGPU

ACPI;Patch; HECI to IMEI

ACPI;Patch; HDAS to HDEF

ACPI;Patch; SAT0 to SAA

ACPI;Patch; LPC to LPCB

ACPI;Patch; PEGP to GFX0

ACPI;QUIRKS;Normalice header; **YES**

ACPI;QUIRKS;rebase regions: **YES**

ACPI;QUIRKS;restHwSig; **YES**

DeviceProperties;Add;PciRoot(0x0)/(Pci(0x1b,0x0);Layout-ID;**01000000**

Entspricht Layout 1 was bei unserem Audiochip funktionieren sollte. (Standard)

DeviceProperties;Add;PciRoot(0x0)/Pci(0x2,0x0);AAPL,ig-platform-id; 0**300923E**

(Hierbei geht es um den Grafikchip; 0300923E wenn wir den internen Grafikchip nicht für die

Ausgabe nutzen)

Kernel;Add;4;IntelMausi.kext; **YES** (aktiviert den Ethernet Treiber)

Kernel;Quirks;AppleXcpmCfgLock; **YES**

Kernel;Quirks;PanicNoKexDump; **YES**

Kernel;Quirks;XhciPortLimit; **YES**

Reine Grafische Sache:

Misc;Boot;ConsoleBehaviourOs; **ForceText**

Misc;Boot;ConsoleBehaviourUi; **Text**

Misc;Boot;ConsoleMode; **Max**

Misc;Boot;Resolution; **1920x1080@32**

Misc;Boot;Timeout; **10**(Wie lange (Sekunden)der Bootloader wartet bis er die erste Auswahloption bootet)

Misc;Security;AllowNvramReset;**YES**

(Damit NVRAM mit bootpicker oder cmd+Opt+P+R resetet werden kann)

Misc;Security;RequireSignature;**NO**

Misc;Security;RequireVault;**NO**

Misc;Security;ScanPolicy;**0**

(Diesen Wert zum ersten Booten setzten, sollte später angepasst werden)

NVRAM;Add;7c436110-AB2A-4BBB-A880-FE41995C9F82;boot-args;**-v keepsyms=1**

(Zeigt beim Start die Konsole was gerate gebootet wird, anstatt des Apple Logos) (Standard)

NVRAM;Add;7c436110-AB2A-4BBB-A880-FE41995C9F82;csr-active-config;**00000000**
(Einstellung für SIP)

(67000000-SIP komplett abschalten; 00000000 komplett einschalten; 30000000 kexte dürfen schreiben)

NVRAM;Add;7c436110-AB2A-4BBB-A880-FE41995C9F82;prev-lang:kbd;**de:3**(Sprache=Deutsch) (Typ auf String umstellen)

PlattformInfo;Automatic; **YES**

UEFI;Protocols;ConsoleControl; **YES**

UEFI;Quirks;IgnoreTextInGraphics; **YES** (wird für Consolenausgabe beim Start benötigt)

UEFI;Quirks;ProvideConcoleGop; **YES** (wird für die Consolenausgabe beim Start benötigt

UEFI;Quirks;SanitiseClearScreen; **YES** (wird für Consolenausgabe beim Start benötigt)

c) Nachdem wir die Datei angepasst haben speichern wir diese und benennen sie in "config.plist" um. (ohne Apostrophe)

INFO: Ihr findet meinen kompletten EFI Ordner im Anhang.

6.) EFI auf den Stick laden

Damit wir diesen Ordner auf den Stick bekommen muss die EFI gemounted werden.

Dazu wird am einfachsten der Clover Configurator verwendet.

Unter „Mounte EFI“ den Stick auswählen und mounten.

Den Ordner öffnen und den **Inhalt** des am Desktop erstellen Opencore Ordners auf das **EFI Laufwerk** kopieren.

ACHTUNG: bearbeitet eure config.plist auf keinen Fall mit dem Clover Configurator, da diese dann nicht mit OpenCore funktioniert

7.) BIOS konfigurieren

Ihr solltet das Bios für die Installation, auf der aktuellsten Version haben (F10).

Wenn ihr noch darunter seid, unbedingt zuvor die Bios Firmware updaten.

Im Bios ändern wir folgende Werte:

MIT ; Advanced Memory Settings ; Extreme Memory Profile (X.M.P) ; **Disabled**

BIOS ; Windows 8/10 Features ; **Other OS**

BIOS ; CSM Support ; **Enabled**

Peripherals ; Initial Display Output ; **PCIe 1 Slot**

Peripherals ; Trusted Computing ; Security Device Support ; **Disable**

Peripherals ; USB Configuration ; Legacy USB Support ;**Enabled**

Peripherals ; USB Configuration ; XHCI Hand-off ; **Enabled**

Peripherals ; Network Stack Configuration ; Network Stack ;**Disabled**

Peripherals ; SATA and RST Configuration ; SATA Mode Selection ; **AHCI**

Chipset ; Vt-d ; **Enabled**

Chipset ; Internal Graphics ; **Disabled** (Wenn ihr die iGPU nicht nutzt)

Chipset ; Audio Controller ; Enabled

Chipset ; Above 4G Decoding → **Disabled**

Power ; Platform Power Management ; **Disabled**

Power ; ErP ; **Disabled**

Power ; CEC 2019 Ready ; **Disabled**

Power ; RC6 ; **Disabled**

8.) OSX Installation

a) USB anstecken und vom USB booten.

b) Anschließend OSX installieren

9.) OSX bootfähig machen

a) Damit euer Hack auch ohne USB-Stick booten kann müssen wir nun die EFI vom Stick auf den PC laden.

b) Hierzu wieder die EFI vom Stick und vom PC mit dem Clover Configurator mounten. Die EFI vom Stick dann einfach auf das EFI-Volumen des PC kopieren.

c) Neustart ohne USB-Stick

d) Euer System sollte laufen.

10.) Feintuning (Nachtrag)

a) Kexte aktivieren
Folgende Kexte sind bereits mit installiert aber Standardmäßig nicht aktiviert:
SMCLightSensor, SMCProcessor, SMCSuperIO

Diese können in der config.plist unter Kernell;Add aktiviert werden.
Hierzu können wir einen bereits existierenden Eintrag eines anderen kext nehmen, diesen Kopieren/Einfügen und entsprechen umbenennen.