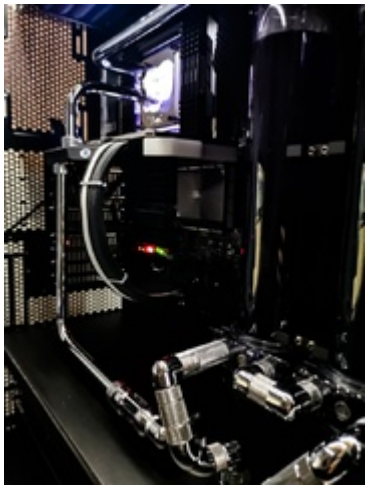
# X299 Tutorial - ASUS WS X299 Sage/10G

### Beitrag von „DSM2“ vom 11. Oktober 2020, 02:13

## X299 Tutorial - ASUS WS X299 SAGE/10G




## Verbaute Hardware:

Gehäuse: Phanteks Enthoo Elite
Gehäuse
Mainboard: Asus WS X299 Sage 10/G
Mainboard
CPU : Intel i9 7960X @4.8 GHz
Intel i9 7960X
RAM: 128 GB Corsair Vengeance LPX 3200 MHz RAM
RAM
GPU: MSI Radeon VII

VII
Gigabyte Titan Ridge Thunderbolt Karte
Titan Ridge
Bluetooth/WiFi : Broadcom BCM943602CS
Broadcom BCM943602CS
Festplatten Intern: 2x Corsair MP510 960 GB

NVMe

Netzteil: Corsair HX1200i
Netzteil

## Mainboard Firmware

Bevor ihr die Installation und die Bios Settings hinterlegt, solltet ihr das Bios des Mainboards Updaten.

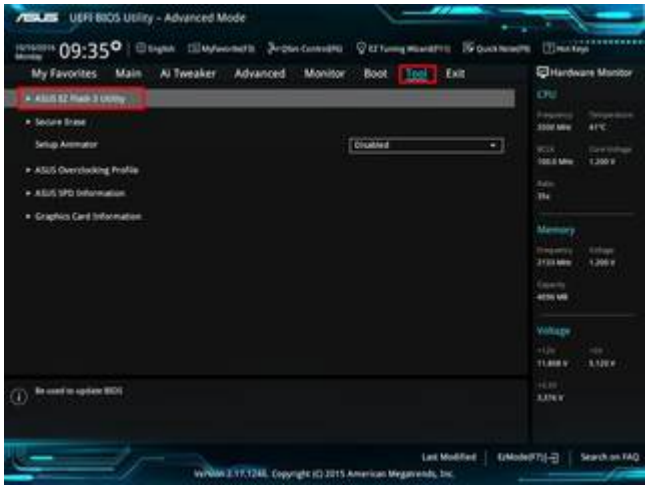
Ladet euch dazu bitte folgendes Bios und extrahiert dieses.

Bios: https://dlcdnets.asus.com/pub/...99-SAGE-10G-ASUS-3203.zip

Anschließend kopiert ihr das Bios File auf einen Fat32 formatierten USB Stick und steckt diesen am Sage 10G an.

Mit der ESC Taste gelangt ihr in das Bios Menü, wo ihr anschließend in das Tool Tab wechselt.

Im Tool Tab angekommen wählt ihr die Option Asus EZ Flash 3 aus und bestätigt diese mit Enter.




Anschließend wählt ihr die Quelle für das Bios File aus, in unserem Fall ist dies der USB Stick und bestätigt erneut mit Enter


Nun wählt ihr das Bios File aus und bestätigt erneut mit Enter,

das Beispiel Bild stammt nicht von einem Asus WS X299 Sage/10G, wie ihr dem unteren Bild entnehmen könnt.

In eurem Fall heißt das Bios File : WS-X299-SAGE-10G-ASUS-3203.CAP


Sobald das Bios Erfolgreich geflashed wurde, könnt ihr die Bios Settings Einstellen.

Doch vorab noch ein wichtiger Hinweis :

Ab Bios Version 3101 benötigt ihr einen AWAK Fix, welcher in der Form einer SSDT hier heruntergeladen werden kann: SSDT-X299-AWAK-Fix.aml

Die SSDT hinterlegt ihr in eurer EFI, andernfalls werdet ihr weder Booten noch macOS Installieren können, sobald ihr die Bios Version 3101 oder nachfolgende nutzt.

Clover User packen die SSDT in das Folgende EFI Verzeichnis: EFI/CLOVER/ACPI/Patched

OpenCore User hinterlegen die SSDT in das Folgende EFI Verzeichnis: EFI/OC/ACPI sowie den nötigen Eintrag in der config.plist damit diese auch geladen wird.

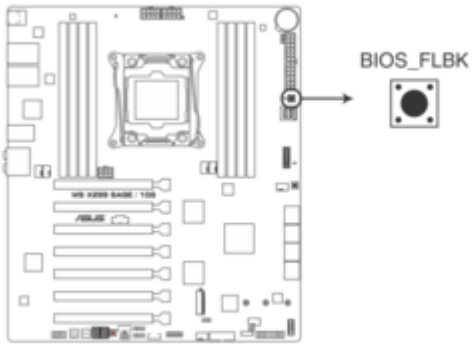
Solltet ihr auf Schwierigkeiten im Flashvorgang treffen oder aber Probleme nachdem Flashen haben, so könnt ihr das Board auch via Flashback Funktion flashen.

Dazu ladet ihr euch das entsprechende Bios File vom Asus Server und benennt das extrahierte CAP File zu WSXTG.CAP um.

Das CAP File hinterlegt ihr anschließend auf einen FAT32 Formatierten USB 2.0 Stick.

Anschließend den Rechner herunterfahren falls dieser in Betrieb war und den USB Stick wie dem Bild zu entnehmen in den Flashback USB Port einstecken:


Der Flashback Button befindet sich hier:



WS X299 SAGE/10G BIOS_FLBK button

Den Flashback Button 3 Sekunden lang festhalten und anschließend loslassen.

Die Flashback LED sollte weiterhin blinken bis der Flashvorgang abgeschlossen ist.

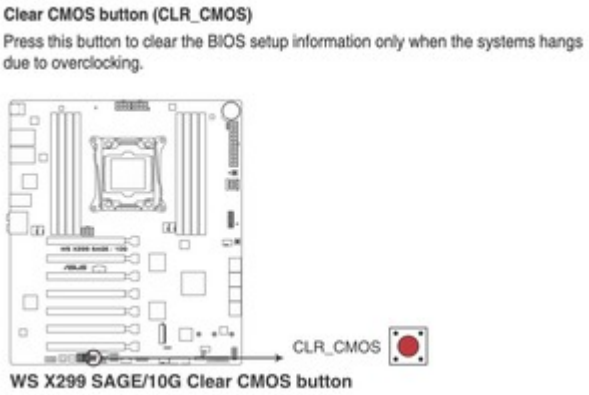
Sollte die Flashback LED nach 3 Sekunden nachdem ihr den Button losgelassen habt mit dem blinken aufhören, so ist entweder der USB Stick nicht dafür geeignet oder aber falsch formatiert.

Es muss sich hierbei um einen USB 2.0 Stick handeln, andernfalls geht das in den meisten fällen schief, weshalb ich euch auch den Einsatz eines USB 3.0 Sticks nicht empfehlen kann, wenn es darum geht das Bios zu flashen.

Die LED erlischt sobald der Flashvorgang abgeschlossen ist von alleine und anschließend könnt ihr das System wieder starten.

Solltet ihr dann nur einen Blackscreen haben dann einmal ein CMOS Reset durchführen, dann sollte alles wieder normal laufen.

Den CMOS Reset Button könnt ihr hier finden:




# Bios Einstellungen

## AI Tweaker

ASUS MultiCore Enhancement: Auto

AVX Instruction Core Ratio Negative Offset: 2

AVX-512 Instruction Core Ratio Negative Offset: 3

CPU SVID Support: Disabled (Sobald man Overclocking betreibt)

## Advanced -> CPU Configuration

Hyper Threading : Enabled

MSR Lock Control : Disabled

## Advanced -> CPU Configuration -> CPU Power Management Configuration

Enhanced Intel Speed Step Technology (EIST): Enabled

Autonomous Core C-States: Enabled

Enhanced Halt State (C1E):Enabled

CPU C6 report: Enabled

Package C State: C6(non retention) state

Intel SpeedShift Technology:Enabled

### Advanced -> System Agent Configuration

Intel VT for Directed I/O (VT-d) : Disabled/Enabled (Falls Enabled dart=0 als Bootflag hinzufügen)

### Advanced -> Thunderbolt (TM) Configuration

TBT Root port Selector : PCIE16_2

Thunderbolt(TM) PCIe Cache-line Size: 128

Security Level: SL0-No Security

### Boot

Fast Boot: Disabled

Above 4G Decoding: On (bei den ganz aktuellen Bios Versionen notwendig da sonst kein Post)

First VGA 4G Decode: Auto

Boot-Reihenfolge einstellen

Boot **->** CSM

Launch CSM: Disabled

Secure Boot **->** OS Type: Other OS

## Erstellen des Bootfähigen USB Sticks für die Installation sowie einer initial EFI

Was wird benötigt ?

1. Zugriff zu einem Computer auf dem Windows/MacOS bereits läuft, egal ob Original Apple Computer/Hackintosh/VirtualBox.

Falls ihr den Weg per VirtualBox gehen wollt, benötigt ihr eine MacOS Lizenz bzw. das Betriebssystem selbst

Original Apple Mac OS X Snow Leopard DVD


Für die VirtualBox Umsetzung gibt es von al6042 ein Video Tutorial wie ihr das ganze installieren könnt.

Ein ganz großes Dankeschön an dieser Stelle!

Tutorial: SL-Virtualbox-on-MJ.m4v.zip

Windows User:

Was wird benötigt ?

1. Big Sur Recovery Image : Recovery Image

2. USB Stick mit mindestens 4 GB Speicher

3. Etcher für die Erstellung des Recovery Sticks : https://github.com/balena-io/e...cher-Portable-1.5.109.exe

## How To

https://youtu.be/YNg9nbYuauE?rel=0&vq=hd1080

Die anderen laden sich MacOS über den **App Store** herunter.

2. Einen USB Stick mit mindestens 8 GB freiem Speicherplatz (bei Catalina 16 GB da der Installer über 8GB groß ist)

3. Zudem benötigt ihr den Clover Bootloader, welchen ihr hier runterladen könnt:

als EFI Zip mit den Clover_v2.5k_r5103.pkg.zip

Nachfolgende Releases können hier bezogen werden: https://github.com/CloverHacky...CloverBootloader/releases

**Bitte beachtet das einige der Treiber in Nachfolgenden Releases mittlerweile nicht mit Clover kommen und daher gar nicht beim Installalieren als Option aufgeführt werden.**

**Diese könnt ihr euch mit dem Kext Updater weiterhin runterladen und müsst diese dann händisch in eure EFI unter EFI/CLOVER/drivers/ einsetzen.**

4. Um die EFI Konfigurieren zu können, benötigen wir zusätzlich Clover Configurator: https://mackie100projects.alte...load-clover-configurator/

## Na dann wollen wir mal...

1. App Store öffnen, Catalina in die Suchleiste eingeben und den Download starten.


2. Sobald der Download abgeschlossen ist, USB Stick einstecken und dessen Bezeichnung zu USB ändern.

3. Terminal starten und je nachdem für welche macOS Version ihr einen Stick erstellen wollt, den entsprechenden Befehl einfügen und mit Enter bestätigen

Code

1. Mojave :
2. 
3. sudo /Applications/Install\ macOS\ Mojave.app/Contents/Resources/createinstallmedia --volume /Volumes/USB/ --nointeraction
4. 
5. Catalina :
6. 
7. sudo /Applications/Install\ macOS\ Catalina.app/Contents/Resources/createinstallmedia --volume /Volumes/USB

Ihr werdet aufgefordert euer Kennwort einzugeben, welches ihr mit Enter bestätigt.

So sieht das ganze dann im Terminal aus, wenn der Stick Erfolgreich erstellt wurde.

4. Clover Installation

https://youtu.be/Tnqt3eeh26Y?rel=0&vq=hd1080

Relevant ist das ihr wie im Video zu sehen, dass ganze im UEFI Mode, sowie in der ESP Partition eures Bootfähigen USB Sticks installiert,

dafür ist "Installation für UEFI Motherboards" und "Installiere Clover in der ESP" angewählt.

Unter UEFI Treiber 64 Bit benötigt ihr : ApfsDriverLoader, FwRunTimeServices, HFSPlus, OcQuirks sowie SMC Helper.

5. Clover Configurator

Da Clover viele Dinge in der Lage ist "On the Fly" zu patchen und je nachdem womit ihr den Bootloader Installiert,

auch gerne mal vorab etwas rein nimmt, was jedoch keinerlei Zusammenhang mit dem X299 Build zu tun hat,

hänge ich hier eine config.plist an welche ihr euch zuerst laden solltet bevor ihr weitermacht.

config.plist

Im Video zeige ich euch was damit zu tun ist, sowie alles was ihr ebenfalls weiter machen müsst.

Ich möchte euch vorab nochmal darauf hinweisen, dass wir zu beginn eine initial EFI erstellen,

womit ihr MacOS installieren könnt und erstmal Zugang zum System habt, mit anderen Worten vieles funktioniert noch gar nicht,

dass ganze Feintuning kommt dann später im Verlauf des Guides.

Da einige Schritte erst mit späteren Versionen hinzukamen,

nachdem ich den Guide geschrieben habe, habe ich einige der Änderungen bereits in der angehängten config.plist hinterlegt.

Daher bitte nicht wundern das ihr da Einträge habt, die ihr in den Videos nicht sehen könnt.

Das hat schon seine Richtigkeit so.

Zum Beispiel habe ich den CPU Type des Intel Xeon W hinterlegt - dient eigentlich nur der Kosmetik aber dennoch.

Ein wichtiger Punkt ist der EC0 zu EC Rename unter ACPI.

Die AppleACPIEC (Bestandteil der AppleACPIPlatform.kext) hätte gerne einen EC Eintrag. (Embedded Controller)

Bei Windows Maschinen heißt dieser jedoch üblicherweise EC0, H_EC oder auch ECDV. (letzteres eher bei Notebooks)

In dem Fall vom Sage 10 G ist der EC Controller als EC0 hinterlegt,

was jedoch dazu führt das ihr ohne Rename gar nicht in den Installer gelangt und bei folgender Ausgabe im Verbose hängen bleibt.

apfs_module_start...

Waiting for Root device...

Waiting on...IOResources...

previous shutdown cause...

Daher habe ich den Rename bereits vorab hinterlegt, damit ihr euch damit erst gar nicht rumschlagen müsst.

Ausserdem beinhaltet die config.plist bereits die Port Limit Patches für Catalina.

Keine Angst, es gibt noch immer genug zu tun für euch!

https://youtu.be/xkE1nZ7QjFo?rel=0&vq=hd1080

In der Boot Section füge ich zunächst 3 Bootflags hinzu.

-v : dieser dient dazu, dass jeder Schritt den MacOS macht als Protokoll gelistet wird

debug=0x100 : verhindert bei einer Kernel Panic den reboot

keepsyms=1 : sorgt dafür das ihr sehen könnt, wo es die Kernel Panic gab.

Optional:

Solltet ihr eine 5700/5700XT euer eigenen nennen, dann muss ausserdem der Bootflag agdpmod=pikera gesetzt werden.

In der Kernel and Kext Patches Section um genau zu sein bei KextsToPatch,

habe ich für euch die aktuellen USB Port Limit Patches die ab 10.14.4 gültig sind bereits vorab hinterlegt,

diese sind am ASUS WS X299 Sage/10G **nicht** notwendig aber der Guide passt auch zu anderen X299 Boards,

daher habe ich den patch aktiv gelassen.

Zudem ist er für die Übersicht besser wenn ihr euch später eine Custom USB Kext erstellen wollt.

Unter SMBios habe ich iMacPro1,1 gesetzt und eine Seriennummer sowie eine SmUUID generiert.

In System Parameters : Custom UUID erstellt

6. Kexte

Damit ein Hackintosh überhaupt booten kann benötigen wir einige Kexte.

Bevor wir die große weite Welt des World Wide Web erforschen, uns den Kopf zerbrechen, wo wir diese den nun runterladen können,

schlage ich vor wir nutzen eine geniale Erfindung von unserem Sascha_77 - Kext Updater

Dieses Tool ist übrigens nicht nur dafür da, um Kexte runterzuladen oder um diese auf dem aktuellen Stand zu halten, sondern es kann deutlich mehr.

Schaut euch einfach mal die Beschreibung bei Gelegenheit an.

https://www.youtube.com/watch?v=UQymf2ALb4A&feature=youtu.be?rel=0&vq=hd1080

Welche Kexte brauchen wir genau und wofür sind sie gut ?

AppleALC - Ton

FakeSMC+Plugins - Ohne FakeSMC wird ein Boot selbst für Chuck Norris unmöglich

Die Plugins die mit FakeSMC dabei sind, sind dazu da, um zusätzliche Parameter auslesen zu können. (CPU Temperatur/GPU Temperatur/Lüfter etc)

Lilu - Ein vielseitiges Plugin das sowohl für AppleALC als auch für WhateverGreen benötigt wird.

TSCAdjustReset - ist ein Kext welches dafür sorgt, dass TSC beim Booten von Mac OS auf einem X299-Board mit Skylake-X CPU richtig Synchronisiert wird.

Diese Kext stammt von interferenc und wurde extra für die X299 Platform geschrieben, ursprünglich war dies mal auf dem VoodooTSCSync aufgebaut,

welcher jedoch beim X299 nicht in der Lage war TSC Synchron zu halten.

Mein 16 Kerner hat 16 echte sowie 16 Virtuelle Kerne (Hyperthreading),

man zählt jedoch von 0 hoch also muss als Wert 31 in der Kext hinterlegt werden.

Falls ihr eine andere X299 CPU nutzt, müsst ihr diesen Wert an eure CPU entsprechend anpassen!

Kext: TSCAdjustReset.kext.zip

USBInjectAll - Damit eure USB Ports funktionieren.

Whatevergreen - Schweizermesser für GPUs, beinhaltet mittlerweile aber auch vielerlei zusätzliche Fixes - benötigt Lilu.

**Falls ihr nicht die von mir angehängte Clover Version nutzt, sondern die Version von Github dann fehlen euch einige Treiber die ihr ebenfalls mit dem Kext Updater laden müsst.**

**In diesem Video seht ihr wie ihr vorgeht und wo diese in der EFI eingesetzt werden müssen.**

**In meinem Fall war die EFI auf dem Desktop und müsste damit diese genutzt werden kann selbstverständlich in die ESP eures Sticks.**

https://www.youtube.com/watch?v=oMeqLk4tFt4?rel=0&vq=hd1080

Damit ist der Bootfähige USB Stick samt Clover fertig und wir können uns endlich der Installation widmen.

## Installation

USB Stick in einen freien USB Port einstecken, falls nicht bereits geschehen und den Computer starten.

Öffnet das Boot Menü mit F8 und wählt den USB Stick aus

Ihr landet im Clover Bootloader, wo ihr nun Boot macOS Install from Install macOS Mojave anwählt und mit Enter bestätigt.

https://www.youtube.com/watch?v=KD4We8eAJbE&feature=youtu.be&rel=0&=&vq=hd1080

macOS ist erfolgreich installiert!

## Feintuning

Zunächst einmal sorgen wir dafür, dass ihr macOS auch ohne Bootstick starten könnt, dafür macht ihr folgende Schritte wie im Video zu sehen.

Ihr mountet zuerst die EFI des USB Stick's und kopiert den EFI Folder auf die ESP Partition eurer

Festplatte.

Stick abziehen und ihr könnt ohne USB Stick booten.

https://youtu.be/b7Bh58p9zNc?rel=0&vq=hd1080

Nachdem wir das gemacht haben, können wir uns endlich dem Feintuning der Config.plist widmen!

## Config.plist abschließend Konfigurieren

Replacement Patches gibt es viele aber nicht alles wird benötigt!

Ich nutze wirklich nur das was für mich einen nutzen hat und sich nicht nur auf optische Ästhetik konzentriert.

Da wir nun die ACPI Tab bearbeiten, solltet ihr unbedingt auch PluginType aktivieren, welches für das Powermanagement relevant ist.

Mountet falls nicht bereit geschehen die ESP eurer Festplatte mit Clover Configurator und öffnet die config.plist um diese bearbeiten zu können.

Kopiert dann entsprechend die im Spoiler angegebenen Patches in die ACPI Tab und sichert eure bearbeitete Config.plist

Patches

Nun wechseln wir in die Devices Section und setzen für die Realtek S1220A die Audio ID 7, sobald diese hinterlegt sichern nicht vergessen.

Nun ist unsere Config.plist für das Asus WS X299 Sage 10/G fertig und theoretisch könnte wir auch schon die drei Bootflags aus der Boot Section wieder entfernen,

da wir aber noch lange nicht fertig sind, schadet es uns nicht, wenn wir eine Möglichkeit zum debuggen haben!

https://www.youtube.com/watch?v=LP1PhGUmXgI&feature=youtu.be?rel=0&vq=hd1080

## Ethernet

Das Asus WS X299 Sage/10G hat 2x 10 Gbit Lan Ports Onboard, dabei handelt es sich um die Intel X550-AT2 die jedoch von Hausaus nicht ohne Nachhilfe rund laufen.

Der beste Weg um das zu beheben und eine Dauerhafte Lösung zu haben, ist es diese per Linux zu patchen und anschließend den Offiziellen Smalltree Treiber zu verwenden.

### Was wird benötigt:

1) USB Stick

2) Windows für das erstellen des Ubuntu USB Sticks via Rufus

3) Rufus - https://rufus.ie/en_IE.html

4) Ubuntu Image - https://www.ubuntu.com/download/desktop

5) SmallTree Treiber für macOS sobald umgesetzt :

High Sierra/Mojave : SmallTreeIntel8259x-3.5.0.dmg.zip

Catalina : SmallTreeIntel8259x-3.8.6.dmg.zip

Windows:

Intel® Ethernet-Controller X550-AT2.zip

## How to:

1) Per Rufus einen USB Stick erstellen

2) Per F8 vom Stick starten und Ubuntu mit ""Try Ubuntu without Installing" booten.

3) Mit dem Internet verbinden

4) Terminal öffnen

und dann kann der Spaß beginnen und ihr kopiert folgende befehle in das Terminal und bestätigt immer mit Enter :

1) sudo apt-get install net-tools

2) sudo apt-get install ethtool

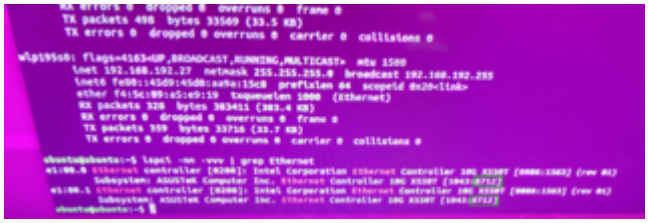
3) ifconfig

Nachdem letzten Command habt ihr dann das vor Augen:

Der Intel X550-AT2 ist enp225s0f0 und enp225s0f1 zugewiesen, diese Adressen können je nach verwendeter Linux Version abweichen.

Wichtig ist das ihr eure enp Adressen entsprechend in den weiter unten folgenden Befehlen anpasst und diese dann für beide Ports ausführt.

4) lspci -nn -vvv | grep Ethernet

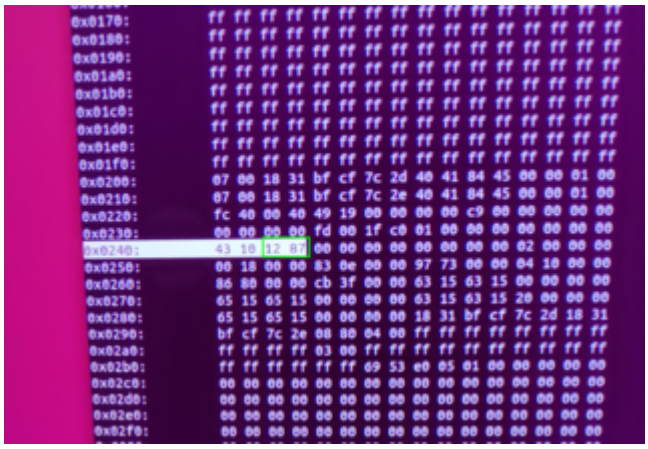


Der Relevante Wert lautet 8712 da dieser zu 000a verändert werden muss auf beiden Ports.

5) sudo ethtool -e enp225s0f0 | less

Es erscheinen eine Menge an Offset Werten aber uns interessiert nur einer: und das ist die Subsystem ID ...

43 10 = 1043 = Subsystem-Hersteller-ID - 12 87 = 8712 = Subsystem-ID !!! Da bist du ja




Folgendes müsst ihr nun ausführen und ich erinnere euch nochmal daran das ihr eure enp abgleicht und falls diese von meinen abweicht entsprechend abändert müsst für beide Ports.

sudo ethtool -E enp225s0f0 magic 0x15638086 offset 0x242 value 0x0a

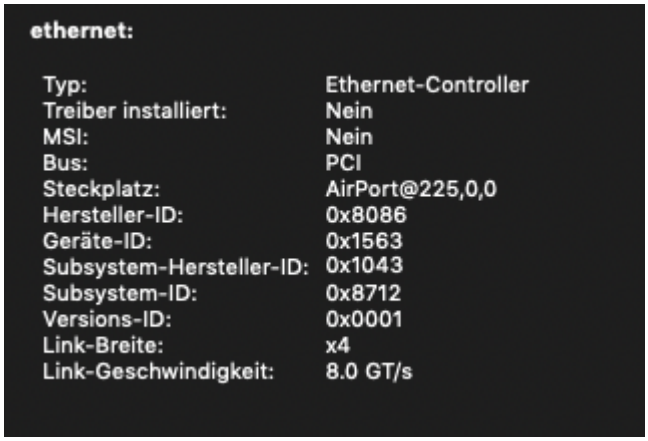
sudo ethtool -E enp225s0f0 magic 0x15638086 offset 0x243 value 0x00

sudo ethtool -E enp225s0f1 magic 0x15638086 offset 0x242 value 0x0a

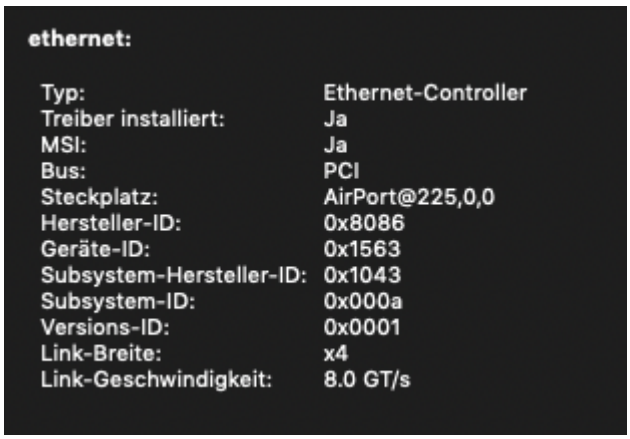
sudo ethtool -E enp225s0f1 magic 0x15638086 offset 0x243 value 0x00

Danach Neustarten und MacOS booten, offiziellen SmallTree 10 Gbit Treiber installieren und sicher in die Zukunft gehen.

Vor dem Patchen:




Nach dem Patchen:




**PS: Ich bin nicht der Erfinder dieser Methode oder habe es rausgefunden...**

**Ich habe es lediglich für meine Intel Karte angepasst bzw umgesetzt und für andere festgehalten,**

**die das vielleicht wiederholen wollen oder müssen.**

**All credits goes to : Squuiid von MacRumors - https://forums.macrumors.com/t...ee-macos-drivers.1968456/**

## PCI Einträge (Kosmetik)

Beispiel Bild

Für das hinzufügen von PCI Einträgen gibt es reichlich Methoden, ich verlinke hier den Guide von Noir0SX welcher einen sehr guten Überblick bietet,

sowie auf die Entsprechenden Methoden anhand von Beispielen eingeht.

Verschieden Methoden um Geräte zu den Systeminformationen - PCI hinzuzufügen

## Thunderbolt

Für mich persönlich eins der wichtigsten Features an meinem Hackintosh.

Thunderbolt war früher mit einigen Hürden verbunden, es funktionierte zwar aber Thunderbolt Devices mussten vor dem anschalten des Computers bereits an sein,

ein Abziehen und wieder anschließen ohne reboot war ebenfalls nicht möglich.

Dank der Arbeit von apfelnico @Crismac2013 kgp-imacpro @LeleTuratti Mork vom Ork @Matthew82 @maleorderbride @nmano sowie @TheRacerMaster ist Hotplug möglich geworden!

Ein großes Dankeschön an dieser Stelle nochmal für euren Einsatz.

Was wird benötigt damit das ganze funktioniert ?

Damit HotPlug funktioniert muss die SSDT samt DTGP Methode angewandt und die entsprechende PCIHotplugCapability per SSDT hinzugefügt werden.

Dafür benötigt ihr diese SSDT : Thunderbolt.zip ( Credits apfelnico )

Dabei ist es wichtig darauf zu achten das die Device Adresse korrekt hinterlegt ist.

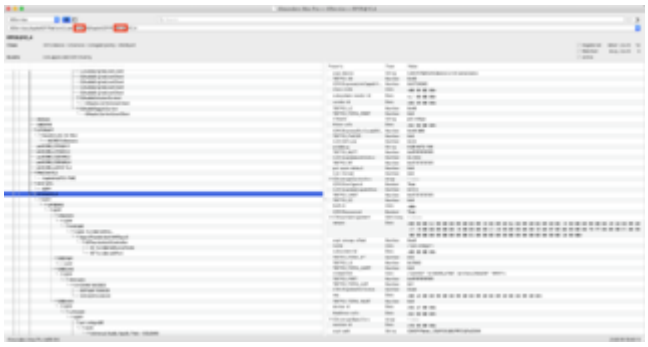
Damit ihr diese korrekt hinterlegen könnt, benötigt ihr IORegistryExplorer sowie MaciASL

**Beispiel am Sage 10/G**

Das ASUS WS X299 Sage/10G hat einen fest zugewiesen PCIe Slot (Slot 2) welcher ausschließlich für Thunderbolt dient.

Thunderbolt hängt in diesem Fall an RP05

(kann bei anderen Mainboards selbstverständlich abweichen)



Öffnet die SSDT-TBOLT3 mit MaciASL und hinterlegt entsprechend die korrekte Adresse in dieser.

Auf die Rot markierten Pfade achten und diese dann entsprechenden in der SSDT hinterlegen.

So sieht das ganze dann aus wenn es fertig ist.

```
External (_SB_.PCI0.RP05.PXSX, DeviceObj)
External (DTGP, MethodObj)    // 5 Arguments
```

**Wie sich jedoch gezeigt hat, kann der Pfad je nach Bios Version abweichen.**

**Bitte daher wirklich kontrollieren ob der Pfad bei euch PCI0 oder zum Beispiel PC00 lautet.**

Speichern und die SSDT in euren EFI/ACPI/Patched Folder einfügen und nach einem Reboot kann der Spaß beginnen.

Gegebenenfalls kann ein NVRAM reset notwendig sein, dafür im Clover Screen einmal F11 drücken,

dass System macht dann einen Reboot bei welchem der NVRAM gelöscht wird.

Anschließend booten

## USB-Custom-Kext

Das ASUS WS X299 Sage 10/G benötigt grundsätzlich keinen Port Limit patch, da das Board nicht mehr als 15 Ports Besitzt,

doch macOS möchte dennoch ganz gerne eine Definition der USB Ports damit diese auch richtig funktionieren und falls notwendig auch intern deklariert werden (Bluetooth).

Damit man die volle Übersicht erhält und die Ports entsprechend funktionieren,

ist der Port Limit Patch sowie der USBInjectAll Kext unumgänglich.

Eine vollständige Übersicht über die Port Limit Patches könnt ihr hier finden : USB Port Limit Patches (Zusammenfassung)

Die aktuellen Port Limit Patches sind in der von mir bereitgestellten config.plist bereits aktiv, anschließend benötigt ihr Hackintool (ehemals Intel FB-Patcher)

Mit dem Tool könnt ihr unter dem Tab USB eine eigene USB Kext erstellen und Ports die nicht zugewiesen (nicht vorhanden) sind entfernen.

Die Vorgehensweise hat unser CMMChris mal super und einfach festgehalten welche ich hier zitiere:

Dann öffnest du Hackintool und wechselst in den USB Tab. Dort siehst du alle USB Ports.

Teste nun der Reihe nach alle deine Ports jeweils mit einem USB 2 und einem USB 3 Gerät durch damit du siehst welche in Benutzung sind. Dann löscht du erstmal die welche nicht genutzt werden aus der Liste.

Danach setzt du die Art des Anschlusses:

- USB 2.0 Anteil eines USB 3 Ports wird auf USB3 gesetzt

- USB 3.0 Anteil eines USB 3 Ports wird auf USB3 gesetzt

- Reine USB 2.0 Anschlüsse auf USB2

- Besonderheit bei Typ-C: Gleicher Port in beide Richtungen = TypeC + SW; unterschiedlicher Port je nach Richtung = TypeC

- Interne USB Ports (z.B. internes Bluetooth) wird auf Internal gesetzt

Sollten deine Anschlüsse das Port Limit von 15 Ports pro Controller sprengen, musst du dich von Ports trennen (1 USB3 Port = 2 USB Ports - USB2 Anteil und USB3 Anteil). Da musst du dich dann selbst entscheiden ob du Anschlüsse komplett deaktivierst, von einem USB 3.0 Port den USB 2.0 Anteil wegnimmst oder umgekehrt.

Sobald alles fertig konfiguriert ist kannst du die Daten exportieren. Hackintool generiert meistens drei Dateien: SSDT-EC, SSDT-UIAC und USBPorts.kext.

- SSDT-EC kommt nach /Clover/ACPI/patched.

- SSDT-UIAC ist für die Verwendung mit USBInjectAll gedacht.

- Die USBPorts.kext ist eine Standalone Lösung, nutzt du diese kannst du USBInjectAll löschen, die SSDT-UIAC brauchst du dann auch nicht.

Wenn ihr dies gemacht habt setzt ihr die USBPorts.kext in eure EFI ein, um genau zu sein in euren EFI/CLOVER/kexts/other Folder und entfernt anschließend den Port Limit patch aus der Config.plist

Für Sage 10/G User hänge ich die Kext an....

USB WS X299 Sage-10G.zip

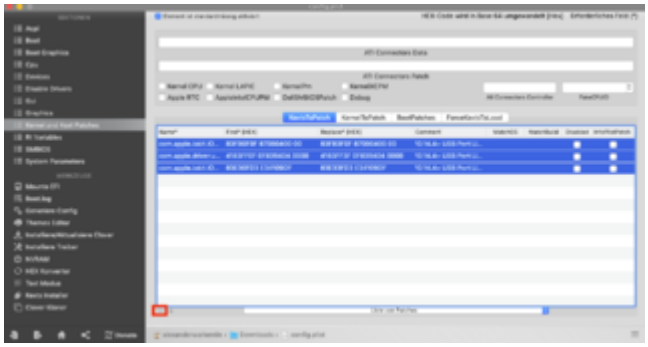
Ich habe vor einiger Zeit mal ein Video erstellt zum hinterlegen der Patches aber dieses eignet sich ebenfalls sehr gut,

um Anhand diesem zu verstehen wo ihr diese wieder entfernen müsst.

https://youtu.be/hJwkoA0bE9g

Zum entfernen wählt ihr die Patches an und entfernt diese mit dem minus Zeichen.

Anschließend Speichern nicht vergessen!



# FERTIG!

## Hilfe und Diskussionen

## ?

X299 Tutorial - ASUS WS X299 SAGE/10G (Hilfe und Diskussionen)