

# Anleitung für Broadcom-WLAN-Patch unter macOS Sonoma

Beitrag von „Huntington“ vom 17. Juli 2024, 14:11

## [Zitat von MacGrummel](#)

Ich glaube kaum, dass der Patch hier ein komplizierter Trick ist. Du musst halt nur ein paar [Kexte](#) zu Deiner PList zufügen, einen [aus dem System ausschließen](#) und den [Patcher](#) drüber laufen lassen. Und vielleicht noch einmal nachsehen, ob die Kexte auch [die aktuellen](#) sind. Sonoma 14.6 [Korrektur [Huntington](#): vermutlich gemeint 14.5] & 14.6 und Sequoia bis zur aktuellen Beta Nr.3 laufen [damit](#) und [hiermit](#) so wie vorher unter Ventura.

Ich bestätige den Hinweis von [MacGrummel](#), dass zur Reaktivierung des WLANs über (nun "legacy") PCIe-Karten von Broadcom (bei mir **BCM94360CD**) auf den momentan neusten Versionen von Sonoma (bei mir **14.5**) die **ALLERneusten Kexts** verwendet werden müssen (**=tl;dr**):

Gestern bin ich von Catalina 10.15.7 auf Ventura 13.6.7 umgestiegen (siehe Hardware links), was lange gedauert hat, weil ich erst [AppleMCEReporterDisabler.kext](#) (auf AMD nötig ab macOS Monterey 12.3) und [AppleIGB.kext](#) ([exakt diese](#) Version - und keine andere! - statt der altgedienten [SmallTreeIntel82576.kext](#) für den I211 Netzwerkkarte meines Motherboards; ebenfalls ab Monterey 12) finden musste. Dabei sind aber auch stabile eigene SSDTs (EC und USBX) mittels SSDTTime sowie eine neue USB Map mittels USBToolBox (auf Windows, Dualboot auf demselben System notwendig) entstanden. Auf Ventura lief soweit alles flüssig, also musste 😊 ich Sonoma probieren.

Da Apple den Support für die Broadcom WLAN-Karten in Sonoma aber fallen gelassen hat, ging die Suche in die zweite Runde... Ich habe mir diesen kompletten Thread durchgelesen, aber es einfach nicht hingekriegt und bestimmt 40 mal mit verschiedenen config.plist-Anpassungen neustarten dürfen. Meines Erachtens hatte ich schon recht früh alles richtig eingestellt - wie es oben wiederholt beschrieben wird:

- OpenCore 1.0.0 mit den neusten Dateien (BOOTx64.efi in Boot/, HfsPlus.efi und OpenRuntime.efi in OC/Driver/, oben erwähnte zwei SSDTs in OC/ACPI/, optionale Tools wie CleanNVRAM.efi, OpenShell.efi und ResetSystem.efi in OC/Tools/),

- die beiden IO-Kexts (bzw. drei Einträge in Kernel/Add mit MinKernel 23.0.0),
- keine spezifischen Kexts für meine BCM94360CD (die keine braucht! - siehe [griven weiter oben](#); z.B. nicht: `AirportBrcmFixup.kext`),
- AMFIPass.kext (damit Firefox etc. funktioniert - siehe [weiter oben; dort folgende](#)) statt `boot-args = amfi=0x80`,
- all diese Kexts in der korrekten Reihenfolge (siehe [weiter oben, @MacGrummel explizit sowie danach](#) und hier unten),
- korrekter Block von `com.apple.iokit.IOSkywalkFamily`,
- SecureBootModel Disabled (kann man scheinbar nach dem Patch mit OpenCore Legacy Patcher wieder zu Default wechseln),
- `csr-active-config = 03080000`.
- Danach Neustart mit NVRAM reset (im Open Core Boot Picker) und Patch mittels OpenCore Legacy Patcher 1.5.0.
  - Auch nützlich [dieser \(deckungsgleiche\) englische Guide und Thread](#).

Gemäss allen Berichten hier sollte das eigentlich klappen. Mein OpenCore EFI war unter (Catalina und) Ventura sowie Sonoma ja stabil (Ruhezustand, Bluetooth, Ethernet, Sound, schneller Boot, alle SATA Festplatten sichtbar - was bei falschen Kexts wie z.B. `AirportBrcmFixup.kext` oder instabilem `AppleIGB.kext` durchaus nicht der Falls ist). Ich sollte alles richtig gemacht haben...?! Aber kein WLAN. **Ich spielte mit dem Gedanken, ob es an der Ethernet-Kext `AppleIGB.kext` liegen könnte - weiter oben wird überlegt, ob `SmallTree` an Kernel Panics beteiligt sein könnte. Nachdem ich es nun geschafft habe, denke ich, dass es unwahrscheinlich ist, dass diese Ethernet-Kexts, sofern sie korrekt und stabil sind, mit der (Dis-)Funktion des WLANs zu tun haben!**

Hier nun mein Beitrag zu dieser Diskussion, der das initiale Zitat von [MacGrummel](#) bestätigt - das Problem war ein anderes: Ich hatte nicht **ALLERneusten Kexts** in meinem EFI. Tatsächlich hatte ich auf meinem **macOS 14.5** (zweitneuste Sonoma-Version) mit SMBIOS MacPro7,1 (spielt wohl nicht wirklich eine Rolle?) `AMFIPass.kext` 1.4.0 statt 1.4.1 und `IOSkywalkFamily.kext` 1.0.0 statt 1.1.0 (siehe Zitat oben). **Diese zu ersetzen genügte jedoch nicht!** Auch die basalsten Kexte waren geringfügig veraltet: **`Lilu.kext` 1.6.7 statt 1.6.8 (vermutlich war es Lilu!!), `VirtualSMC.kext` 1.3.2 statt 1.3.3 (vielleicht auch relevant?), `WhateverGreen.kext` 1.6.6 statt 1.6.7, `AppleALC.kext` 1.9.0 statt 1.9.1, `NVMeFix.kext` 1.1.1 statt 1.1.2, `RestrictEvents.kext` 1.1.3 statt 1.1.4.**

**Es sei dabei zu beachten, dass die jeweils erstgenannten die momentan aktuellen auf in den entsprechenden Repositories auf GitHub unter "Releases" verfügbaren Versionen ("latest") sind! Die von mir jeweils zweitgenannten ALLERneusten Versionen kriegt man hingegen (vereinfacht) hier: <https://dortania.github.io/builds/> !!! (Hackintool kann das ggf. auch...)**

Nach dem Ersetzen dieser Versionen, der zugehörigen Aktualisierung meines config.plist und einem Reboot funktioniert mein WLAN via meine BCM94360CD nun einwandfrei! 😄 Ein erneutes Patchen mittels OpenCore Legacy Patcher war dabei nicht notwendig.

Die Kextreihenfolge und -versionen sind bei mir auf macOS 14.5 folgende:

- Lilu 1.6.8 >> VirtualSMC 1.3.3 >> WhateverGreen 1.6.7 >> AppleALC 1.9.1 (allesamt Schlüssel-Kexts)
- >> AppleMCEReporterDisabler.kext 1.2 (nur AMD ab macOS 12.3)
- >> AMFIIPass.kext 1.4.1 >> IOSkywalkFamily.kext 1.1.0 >> IO80211FamilyLegacy.kext 1.0.0 >> AirPortBrcmNIC (in vorigem enthalten, 1.1.0) (siehe den Guide in diesem Thread)
- >> AppleIGB.kext (spezifisch für I211 Netzwerkkarten ab macOS 12, exakt die oben referenzierte Datei; ggf. 5.11.4?)
- >> USBToolBox 1.1.1 >> die zugehörige eigene UTBMap (USB Map; beide mittels via USBToolBox erstellt)
- >> RestrictEvents 1.1.4 (um irrelevante, störende Fehlermeldungen unter SMBIOS MacPro7,1 zu unterbinden und Updates zu ermöglichen - siehe für inkrementelle Updates [Arkturus weiter oben!](#))
- >> und zuletzt NVMeFix 1.1.2 (nicht notwendig bei meiner Hardware, hilft aber vielleicht beim Energiemanagement meiner SSD).
  - OCAuxiliaryTools können dabei helfen, die Version der Kexts in einem config.plist zu erkennen (bei IO80211FamilyLegacy etwas verwirrend). ~~Dieses Programm ist momentan scheinbar aber nur bis OpenCore 0.8.8 (!) verfügbar. Man kann die Kext-Versionen damit einsehen, sollte aber also keine Konfiguration damit vornehmen!~~

*Ich danke allen sich hier so fleissig austauschenden Personen für die beeindruckende, grosse Arbeit! Hoffentlich kann auch mein Beitrag jemandem helfen.*