

**Erledigt**

## **(Anleitung) Monterey & Big Sur auf Dell Latitude e5450/e7450/e7250/e5250/e5550 + Touchscreen + OpenCore + 100% Tastaturbelegung**

**Beitrag von „Retch“ vom 8. November 2019, 22:22**

Hallo zusammen,

da ich es nun nach etlichen Stunden geschafft habe mein e5450 mit Touchscreen und Trackpad zum laufen zu bringen, wollte ich meine Ergebnisse teilen.

Das EFI sollte grundsätzlich für die Lattitudes funktionieren, die eine Broadwell CPU verbaut haben.

Bei dem e5450 handelt es sich um einen Business Laptop von Dell der aktuell für 170-230€ gebraucht zu bekommen ist. Dieser Laptop hat mein t420s abgelöst und mit dieser funktionierenden Konfiguration bin ich sehr glücklich damit.

Es gibt diese Modelle (eher selten) auch mit Touchscreen.

Zudem müsste eventuell die USBPorts.kext individuell angepasst werden, aber zumindest bei dem e5450 sollte die ohne Probleme ihren Dienst tun.

Bis auf die USB Ports sollte die Efi -wie erwähnt- mit den Broadwell Modellen funktionieren.

Die Seriennummer etc muss natürlich selbst generiert werden, am besten mit [GenSMBios](#) als MacBookPro12,1



### **Erstmal zum Laptop, bzw was alles funktioniert:**

- i5 5300u
- 1080p Display
- ELAN Touchscreen (USB, 10 Finger Multitouch)
- ALPS V7 Touchpad
- O2 Micro SD Card Reader
- Trackpad & Trackpoint
- DW1830 / 1560 BT & Wifi
- Audio ALC3235 (ALC293)
- Boot Menu mit F12
- alle Tasten auf der Tastatur funktionieren nun wie in Windows
- Update auf BIOS A24 läuft problemlos unter MacOS
- DSDT für BIOS A6 & A13. Es sind im EFI Ordner beide vorhanden. Einfach die passende DSDT in der config.plist unter ACPI/Add aktivieren

*//Die Nvidia Grafik und der Fingerprint Reader funktionieren wie gewöhnlich nicht unter Mac OS*

//Die SSDT-Disable-DGPU.aml deaktiviert die Nvidia Grafik, wenn keine verbaut ist kann man diese in der config.plist einfach deaktivieren.

## Touchscreen

Der Touchscreen hat bis Mojave/Catalina Installationsassistent ohne kext funktioniert, aber seit dem installierten Catalina fehlt die Unterstützung dafür.

Im Hackintool wird er als USB Gerät Gerät erkannt. Das Programm [ControllerMate](#) erkennt den Touchscreen ebenso und kann die Inputs in z.B. eine Virtuelle Maus umleiten.

Eine Config dafür habe ich erstellt, wenn ich weitere Gesten hinzufüge wird die Config selbstverständlich aktualisiert. [ControllerMate](#) lässt sich kostenlos nutzen, ich bin noch nicht an die Free-Limitationen gestoßen.

Den Touchscreen im laufenden Betrieb könnt ihr [HIER](#) sehen.

Die Touchscreen Gesten wie Mission Control oder Launchpad benötigen [Karabiner](#) um zu funktionieren, da ControllerMate die Launchpad Geste an F9 weitergibt und Mission Control über F8 läuft. Hört sich zwar umständlich an, funktioniert aber absolut reibungslos.

ControllerMate sollte generell für USB Touchscreens geeignet sein, die Config kann man dann für seinen Laptop anpassen.

ControllerMate bekommt es nicht hin, seinen eigenen Export zu importieren, es kommt dann immer die Meldung "...enthält keine Daten".

Das soll uns aber nicht daran hindern, die `Programming.plist` welche sich in `~/Library/Application\ Support/ControllerMate/` befindet einfach mit meiner `Programming.plist` zu ersetzen. Entweder macht man das manuell, oder man führt das `meconfig.sh` Script im Terminal aus, was das erledigt. In der `ControllerMateConfigImport.zip` ist die `Programming.plist` sowie das kleine Bash Script enthalten.

Dank [lumealex](#) Hinweis funktioniert der Touchscreen nun ohne Zutun als "Trackpad". ControllerMate wäre ab Big Sur sowieso Geschichte gewesen da es scheinbar keine Updates mehr bekommt.

## Tastatur

Wo ich schon bei [Karabiner](#) bin, kann ich auch gleich die Konfiguration der Tastatur erklären.

Mich hat es gestört, dass beim Coden die Klammern nicht so belegt sind, wie auf den Tasten

gelabelt. Habe wieder etliche Stunden gebraucht um die richtige Lösung zu finden. Das Beste dabei ist, dass sich alles über [Karabiner](#) konfigurieren lässt. Damit man sich nicht mit der [karabiner.json](#) Config beschäftigen muss, welche bei mir mittlerweile über 500 Zeilen groß ist, gibt es ein kleines Helfertool namens [Goku](#). Die Funktionsweise wird auf dem verlinkten [Github](#) beschrieben. Goku lässt sich, wenn man sich eingelese hat, relativ einfach konfigurieren. Die knapp 60 Zeilen der [karabiner.edn](#) (für Goku) sind jedenfalls eine andere Hausnummer als die 500 der [karabiner.json](#).

Das wars erstmal zur Konfiguration von Goku, das müsst ihr ja eh nicht mehr machen da es bereits ausreicht die durch Goku gepatchte [karabiner.json](#) aus dem Anhang herunterzuladen und einzufügen. Wer dann selber noch konfigurieren möchte sollte sich näher mit Goku beschäftigen.

Die [karabiner.json](#) aus dem Anhang gehört dann in `~/.config/karabiner/` geschoben. In dieser Config sind auch gleich die fn-Tasten belegt.

Die [karabiner.edn](#) gehört in `~/.config/`, danach den Befehl `goku` ausführen. //Nur nötig wenn man selber die Tasten konfigurieren möchte

Im Anhang gibt es noch nützliche Bilder die für die Konfiguration nützlich sein könnten.

Hinweis: Die Tasten `Alt` und `Alt Gr` haben nun die selbe Funktion. `@` funktioniert über `Alt(Gr)` und `Q`, also so wie gelabelt.

Wer zusätzlich das Zeichen '≠' benötigt, kann es mit `Alt(Gr)+6` einfügen. [Direktlink](#)

Ich habe versucht die [karabiner.edn](#) so verständlich wie möglich zu kommentieren, wenn es Probleme geben sollte, einfach einen Beitrag hier drunter setzen.

Die Anleitung lässt sich natürlich auch für Desktops verwenden, hier kann es allerdings sein, dass gewisse Tasten vertauscht sind.

[Hier](#) ist noch eine Config meiner Logitech G413 zu finden, sollte für Desktop PC's eventuell eher geeignet sein. Einfach mal probieren welche funktioniert und im Zweifel halt selber verbessern.

PS: In den beiden Zip Archiven sind jeweils eine [karabiner.edn](#) sowie die gepaschte [karabiner.json](#) zu finden.

Hinweis für Nutzer von Bios Version niedriger als A07: Wenn es vorkommt, dass eine Taste nach dem Drücken immer noch aktiviert ist, sollte dieser Fehler mit dem Bios A07 behoben sein.

## Touchpad

Angepasst ist ebenso die  
VoodooPS2ControllerR7A1dsarch.kext/VoodooPS2Trackpad.kext/Info.plist.

~~Bei nicht-Full HD muss hier eventuell auch noch die Resolution geändert werden.~~

## OpenCore

0.7.5 läuft jetzt auch auf dem Laptop.

Der Clover Ordner wird keine Updates mehr bekommen - bitte den Opencore Ordner nutzen.

## DSDT/SSDT

Damit alles richtig funktioniert (insbesondere Probleme beim Starten/Sleep/Display), war unter Clover der Haken "Fix HPET" notwendig.

FixHPET

~~Unter Opencore braucht es dazu eine DSDT, in welcher der "IRQ Fix" aus dem Rehabman-Laptop Repository angewandt wurde.~~

~~Ich nutze die Bios Version 6, wenn man also die hat kann man beim e5450 die DSDT einfach verwenden. Für Die Bios Version 13 gibt es [hier](#) die passende DSDT zum herunterladen. Es sind im EFI Ordner beide vorhanden. Einfach die passende DSDT in der config.plist unter ACPI/Add aktivieren.~~

~~Bei anderen Laptops bzw anderen Bios Versionen muss der IRQ Fix selber angewandt werden.~~

Keine DSDT mehr notwendig. Das funktioniert jetzt über die SSDT-HPET und drei Patches, es muss nichts mehr gemacht werden. Dafür war [SSDTime](#) sehr hilfreich.

## Downloads

Es werden nicht alle Anhänge angezeigt, deshalb hier die Direktlinks:

Opencore Configs bei [Github](#)

[karabiner configs G413 with Alt+6 ≠.zip](#)

[karabiner configs e5450 with Alt6 ≠.zip](#)

## **BIOS Settings**

Boot Sequence -> Boot List Option -> UEFI

Advanced Boot Options -> Enable Legacy Option ROMs -> an

SATA Operation -> AHCI

TPM Security -> aus

Computrace -> Deactivate

CPU XD Support -> an

Secure Boot Enable -> Disabled

## **Lesestoff**

<https://github.com/yqrashawn/GokuRakuJoudo>

<https://github.com/yqrashawn/G...o/blob/master/tutorial.md>

<https://github.com/pqrs-org/Karabiner-Elements/issues/925>

## **Schlusswort**

Letztendlich kann ich sagen, dass das einiges an Arbeit war und ich nun ein Hackbook habe, welches sich tatsächlich wie ein Macbook verwenden lässt und man nicht auf "Sonderzeichensuche" gehen muss.

Das wars erstmal meinerseits, weitere Informationen und Updates werde ich von Zeit zu Zeit ergänzen.

## **Updates:**

- Tastaturkonfiguration hinzugefügt

~~- ControllerMate kann seine eigene .cmate Konfiguration nicht importieren, deshalb bitte die ControllerMateConfigImport.zip benutzen~~

~~- Clover jetzt auf Version 5103~~

- ~~- Touchscreen kann jetzt Launchpad, Mission Control und Notification Center~~
- CPU Powermanagement hinzugefügt
- OpenCore Support 0.6.2
- Touchscreen wird nun als Trackpad erkannt, so braucht man ControllerMate nicht mehr
- keine DSDT mehr notwendig