

Anleitung: Installation und Inbetriebnahme Yosemite auf einem Acer Aspire VN7-791G Teil 1

Beitrag von „mc_killa“ vom 31. Mai 2015, 20:10

Hallo Zusammen,

Die verbaute Hardware (In diesem Fall von meinem VN7-791G):

Chipsatz: Intel Haswell HM86

CPU: Intel i7-4720HQ / 4x 2,60 GHz

Ram: 2x 8 GB DDR3-1600

Grafikkarte: Intel HD4600 / nVidia GeForce GTX960M

Sound: Realtek ALC283

Festplatte: 256 GB M.2 SSD / 1 TB Sata HDD 5400 RPM

Optisches Laufwerk: Blu-Ray Brenner Matshita UJ272Q ATA

LAN-Karte: Gigabit Ethernet Broadcom Netlink BCM57780

W-LAN Karte: Qualcomm Atheros QCA6174 inkl. Bluetooth

USB: 2x USB 2.0 / 2x USB 3.0 / Cardreader als USB 2.0 CRW

Sollte für den Anfang als Info reichen.

Folgende Dinge funktionieren nicht, da inkompatibel:

W-LAN Karte: Qualcomm Atheros QCA6174 inkl. Bluetooth

Diese funktioniert nicht, da es noch keinen Treiber hierfür gibt. Am Besten (So hab ich es gemacht) kauft ihr euch einen USB-WLAN-Dongle oder nutzt halt nur das Kabelgebundene Netzwerk.

Grafikkarte: nVidia GeForce GTX960M

Meines wissens existiert keine möglichkeit nVidia Grafikkarten zusammen mit der Internen Grafik zu nutzen. Da ich es für mich nicht funktionsfähig gebracht habe, bzw es gar nicht erst versucht habe, werde ich hierzu einfach nur sagen, es wird "wahrscheinlich" nicht funktionieren.

Beginnen wir:

1. Installation von Yosemite:

Jeder nutzt einen anderen Bootloader, daher kann ich hier nur meine Methode angeben: Ich habe mittels Virtueller Maschine bereits ein Apple OS installiert und habe dort mit qlover Note gearbeitet, um einen USB-Stick für Yosemite zu erstellen.

Google bietet hierzu reichlich Anleitungen, daher gibt es hier keine Erklärung meinerseits (Ganz abgesehen davon ist das Programm sehr selbsterklärend).

Wenn ihr den Stick erstellt habt, geht bitte auf die EFI Partition eures USB-Sticks (Wird automatisch im System gemounted sobald der Stick erstellt wurde) dann auf den Ordner Clover und auf die config.plist.

Es ist nämlich so, die Standard config.plist funktioniert nicht bei dem VN7 (Bei installationsstart werden Grafikfehler erzeugt, die dafür sorgen, dass man beim Hochfahren des Setups nichts mehr machen kann) als erstes solltet ihr euch also eine ordentliche config.plist zulegen, ich habe eine von RehabMan genutzt auf dieser Seite: <https://github.com/RehabMan/OS-X-Clover-Laptop-Config>

Hier habe ich die config.plist der [config_HD4600_4400_4200.plist](#) genutzt. Einfach herunterladen und in config.plist umbenennen und in den Clover Ordner kopieren. Die darin enthaltene config.plist vorher entfernen oder überschreiben lassen.

Aber bevor ihr es jetzt versucht, ändert noch folgende Zeile in der config.plist

Zitat

```
<key>#ScreenResolution</key>
<string>1920x1080</string>
<key>Theme</key>
<string>Bluemac</string>
```

und zwar ändert ihr in der Zeile die 192x1080 auf 1024x768 und entfernt das Hashtac vor der "ScreenResolution". Sollte dann so aussehen:

Zitat

```
<key>ScreenResolution</key>
<string>1028x768</string>
<key>Theme</key>
<string>Bluemac</string>
```

Somit wird eine Manuelle Auflösung gewählt von 1024x768. (Sieht zwar extrem aus am Anfang aber es tauchen keine Bildfehler mehr auf)

Nun könnt ihr Yosemite installieren und wir können mit den nächsten Schritten fortfahren.

2. Native DSDT extrahieren:

Viele wissen es ja bereits, die DSDT kann man auf vielerlei Art und Weise extrahieren. Für die, die es nicht wissen, ist hier eine kleine Übersicht:

Mit Windows: könnt ihr die DSDT (ACPI Tabellen) mit AIDA64 Engineer auslesen.

Dazu einfach das Programm öffnen und dann oben in der Leiste den Reiter "Werkzeuge" anklicken und auf ACPI Browser gehen.

Es erscheint eine Tabelle wo ihr eure DSDT und SSDT's finden könnt. Unten seht ihr dann auch den Button "Save DSDT". Wenn ihr diesen drückt, fragt er euch direkt wo die dsdt hin soll. (Hier könnt ihr einen Ort eurer Wahl wählen) **Bitte beachten:** Dies ist die DSDT in einer *.bin Datei also ein Binäre Datei. Die muss erst ins *.aml Format gebracht werden (viele benennen diese einfach um). Da ich diese Methode mittels Windows nicht empfehle werde ich hier nicht weiter drauf eingehen. 😊

Mit Linux: Meine bevorzugte Methode. Ich habe hierfür Ubuntu in der Version 14.04.2 Heruntergeladen (Ist Gratis) und noch das Programm LinuxLive USB Creator 2.9.2. So könnt ihr euch Kinderleicht einen Linux Live USB-Stick erstellen und müsst Linux nicht auf eurem Computer/Laptop installieren. Startet Ubuntu/Linux im Live-Modus und öffnet das Terminal.

Im Terminal gebt ihr dann folgenden Befehl ein:

Zitat

```
# substitute DEST with the mountpoint of a FAT32 formatted USB stick  
sudo cp -R /sys/firmware/acpi/tables DEST
```

Dieser Befehl extrahiert die Nativen ACPI Tabellen in den von euch angegebenen Ordner und zwar direkt im *.aml Format.

Diese Methode bevorzuge ich, weil sie Simpel, Schnell und Effizient ist, da direkt alle wichtigen Dateien die ihr braucht im richtigen Format da sind.

3. *.aml Dateien in das Format *.dsl bringen (Disassembeln) 😞

Um mit den Dateien arbeiten zu können (um sie zu patchen), müssen die *.aml Dateien ins *.dsl Format gebracht werden, hierzu einfach das Programm iasl nutzen, welche ihr hier findet:

<https://bitbucket.org/RehabMan/acpica/downloads>

Dieses Programm entpackt ihr und kopiert die darin enthaltene iasl mittels Terminal und folgendem Befehl in eure OSX /usr/bin

Zitat

```
sudo cp -R /Ordner/der/iasl/Datei /usr/bin/
```

Jetzt befindet sich iasl in eurem OSX System und ihr könnt im Terminal mit folgendem Befehl aus den *.aml Dateien, die Dateien erstellen, die ihr zum Patchen braucht:

Zitat

```
cd /Ordner/wo/sich/eure/*.aml/Dateien/befinden/  
iasl -da -dl *.aml
```

Nun sollte im selben Ordner wo die *.aml Dateien sind, auch die *.dsl Dateien sein und wir können nun mit der Einstellung für unser System beginnen.

(Dies war nur eine Kurze Anleitung für die DSDT/SSDT Erstellung, wer alles erfahren möchte kann sich [hier](#) schlau lesen. 😊)

4. DSDT/SSDT Dateien in Clover einfügen und erste Komponenten lauffähig machen:

Hier benötigen wir erstmal das Programm Maciasl, damit könnt ihr die *.dsl Dateien öffnen und auch bearbeiten.

Ihr

solltet bevor ihr Patche in die DSDT/SSDT's einfügt, erstmal mit Maciasl und dem Compile Button kontrollieren, ob Syntax Error vorhanden sind. Sollten welche vorkommen (Bei mir waren einige) behebt diese bitte zuerst. (Ich werde hier nicht jeden einzelnen Syntax Error erklären, da sonst die Anleitung irre lang wird, Google bietet dazu reichlich Hilfe).

Sobald ihr die Fehler behoben habt, speichert die *.dsl Datei, drückt Compile und Speichert die Datei als "Save as" *.aml (ihr wählt im Speichern als Fenster einfach in der unteren Leiste "ACPI Machine Language Binary" aus und drückt Speichern. Nun habt ihr eine fehlerfreie *.aml und könnt diese problemlos verwenden.

Beachten: Es

ist nicht immer so, dass die DSDT alle Komponenten verwaltet, die euer Notebook hat. Z.b. befindet sich die Intel HD 4600 bei mir in der SSDT-4. Darauf komme ich aber zurück wenn wir ans Patchen gehen, für euch nur zur Info, dass ihr nicht nur die DSDT fehlerfrei habt, sondern auch wenns geht **alle** SSDT's.

Kopiert nun bitte alle

fehlerfreien DSDT/SSDT's in Clover (/Efi/Clover/ACPI/patched/) in der Clover config.plist wäre es ganz gut, wenn diese mit diesem Abschnitt beginnt:

Zitat

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>ACPI</key>
<dict>
<key>DSDT</key>
<dict>
<key>Debug</key>
<false/>
<key>DropOEM_DSM</key>
<false/>
<key>Fixes</key>
<dict>
<key>FixRegions_10000000</key>
<true/>
<key>NewWay_80000000</key>
<true/>
</dict>
<key>ReuseFFFF</key>
<false/>
</dict>
```

```
<key>DropTables</key>
<array>
<dict>
<key>Signature</key>
<string>DMAR</string>
</dict>
</array>
<key>SSDT</key>
<dict>
<key>DropOem</key>
<true/>
<key>Generate</key>
<false/>
<key>PluginType</key>
<integer>1</integer>
</dict>
<key>SortedOrder</key>
<array>
<string>SSDT.aml</string>
<string>SSDT-1.aml</string>
<string>SSDT-2.aml</string>
<string>SSDT-3.aml</string>
<string>SSDT-4.aml</string>
<string>SSDT-5.aml</string>
<string>SSDT-6.aml</string>
<string>SSDT-7.aml</string>
<string>SSDT-8.aml</string>
</array>
</dict>
```

Alles anzeigen

Dies

ist ein Beispiel meiner config.plist. Damit werden die SSDT's und die DSDT sorgfältig in Clover eingebunden. Aus dem Ordner /Efi/Clover/Kexts/ könnt ihr, bis auf den Ordner "Other" alles löschen. In dem nun verbliebenen Ordner "Other" löscht ihr nun alles, bis auf VoodooPS2Controller.kext und FakeSMC.kext (Da für die Zukunft nichts weiteres benötigt wird)

Wenn ihr nun neu startet, könnte es sein, dass fast alle Komponenten bereits erkannt werden. (Könnte, muss aber nicht).

Beitrag von „mc_killa“ vom 31. Mai 2015, 20:14

Nun

bringen wir die Lan-Karte zum laufen, hierzu benötigen wir einen kext, der normalerweise nur für Mavericks geeignet ist, aber der auch bei Yosemite ausgezeichnet funktioniert. Diesen bekommt ihr [HIER](#).

Dieser

Kext wird in einen anderen Kext eingebunden und zwar geht ihr dazu in den Ordner /System/Library/Extensions/ (/S/L/E/) und sucht die Datei "IONetworkFamily.kext" Rechtsklick auf "Paketinhalt Anzeigen" dann auf den Ordner "Contents" und auf den Ordner "Plugins". Da kopiert ihr nun die BCM5722D.kext hinein. Danach natürlich noch die Zugriffsrechte reparieren (Am besten klappt dies mit dem Programm "Kext Utility" ihr könnt es aber auch über das Festplattendienstprogramm machen) danach neu starten und euer Kabelgebundenes Netzwerk sollte funktion haben.

So, nun haben wir Internet und können unter anderem Maciasl gezielter zum Patchen nutzen.

Edit: Die kext muss nicht unbedingt in der IONetworkFamily.kext eingebunden werden, nach meinem aktuellsten Test, geht es auch wenn ihr die Kext mit Kext Utility nur nach /S/L/E/ installiert.

Es

sind Standardpatche in Maciasl enthalten, diese habe ich nicht verwendet, ich habe ausschließlich die Patches von RehabMan genutzt und zwar hier: <https://github.com/RehabMan/Laptop-DSDT-Patch>

Als

erste Patchen wir die Grafik: hierzu sucht eure SSDT, wo die Grafik enthalten ist (In Maciasl links im Device Baum mal alles aufklappen und nach gfx0 suchen) sobald ihr die richtige gefunden habt, geht ihr oben auf "Patch" und auf "open". Sucht euch aus den Laptop DSDT Patch Ordner unter Graphics die [graphics Rename-GFX0.txt](#)

Datei raus und drückt unten auf "open" nun solltet ihr unten im Fenster sehen, dass er einige Changes macht, sobald ihr den Patch mit "apply" bestätigt habt, geht ihr wieder auf "open" [graphics HD4600 yosemite.txt](#) Datei raus und sagt ihm "open". Wenn er hier keine Changes vornimmt (man sieht es ja unten) dann habt ihr die

falsche SSDT. 😊

Wenn

ihr diesen mit "apply" bestätigt und keine Fehler habt, dann habt ihr eure Grafik so gut wie fertig eingerichtet. Aber halt noch nicht ganz.

Nun integrieren wir die Grafikkarte in den Treiberordner. Dafür brauchen wir FakePCIID.

Dieses findet ihr hier: <https://bitbucket.org/RehabMan/os-x-fake-pci-id/downloads>

In

diesem Paket, im Ordner "Release" findet ihr die FakePCIID.kext und die FakePCIID_HD4600_HD4400.kext. Diese beiden Kexte werden mittels Kext Utility in /S/L/E/ installiert (oder Manuell und dann die Zugriffsrechte reparieren). Nun wird eure Grafikkarte beim Neustart komplett mit QE/CI Unterstützung erkannt. (Nach diesem Schritt, kann die "ScreenResolution" von Clover (Siehe Punkt 1) wieder auf Ursprung gesetzt werden. Fügt nur, um eventuellen Grafikfehlern vorzubeugen, folgenden Abschnitt hinzu:

Zitat

```
<key>KextsToPatch</key>
<array>
<dict>
<key>Comment</key>
<string>Boot graphics glitch, 10.10.0/10.10.1 (1 of 2)</string>
<key>Name</key>
<string>#IOGraphicsFamily</string>
<key>Find</key>
<data>hcB0XUg=</data>
<key>Replace</key>
<data>McB0W0g=</data>
</dict>
<dict>
<key>Comment</key>
<string>Boot graphics glitch, 10.10.0/10.10.1 (2 of 2)</string>
<key>Name</key>
<string>#IOGraphicsFamily</string>
<key>Find</key>
<data>QYjE6wM=</data>
<key>Replace</key>
<data>QYjE6yM=</data>
</dict>
```



```
<dict>
<key>Comment</key>
<string>Boot graphics glitch, 10.10.2 (1 of 2)</string>
<key>Name</key>
<string>IOGraphicsFamily</string>
<key>Find</key>
<data>QYjE6xE=</data>
<key>Replace</key>
<data>QYjE6zE=</data>
</dict>
<dict>
<key>Comment</key>
<string>Boot graphics glitch, 10.10.2 (2 of 2) (seems to have no effect)</string>
<key>Name</key>
<string>#IOGraphicsFamily</string>
<key>Find</key>
<data>hcB0a0g=</data>
<key>Replace</key>
<data>McB0W0g=</data>
</dict>
```

Alles anzeigen

Somit tauchen keinerlei Bildstörungen mehr auf.

Nun kümmern wir uns um die Soundkarte.

Dazu fügen wir in unserer DSDT einen Patch von RehabMan's Repo ein, und zwar unter Audio die Datei audio_HDEF-layout12.txt.

Wenn

ihr diesen Patch öffnet, müsst ihr oben in der Zeile " "layout-id",
Buffer() { 12, 0x00, 0x00, 0x00 },\n" die 12 auf die Nummer 86 ändern.

Dann "apply" und fertig. Dann noch ein kleiner Feinschliff, im Ordner
"system" den Patch "system_IRQ.txt zu integrieren, damit eventuelle
Fehler ausgeschlossen sind!

Nun brauchen wir folgenden Ordner: <https://github.com/RehabMan/Lenovo-U430-Touch-DSDT-Patch>

Die in diesem Ordner enthaltene Datei "AppleHDA_ALC283.kext muss mit Kextutility
zusammen mit dieser Kext <https://bitbucket.org/RehabMan...codec-commander/downloads>
installiert werden.

Nun muss das ganze noch ins Clover

integriert werden, hierzu bitte folgendes in die config.plist einfügen:

Zitat

```
<key>KextsToPatch</key>
<array>
<dict>
<key>Comment</key>
<string>AppleHDA ALC283 (1 of 3)</string>
<key>Name</key>
<string>AppleHDA</string>
<key>Find</key>
<data>PYQI7BA=</data>
<key>Replace</key>
<data>PQAAAAA=</data>
</dict>
<dict>
<key>Comment</key>
<string>AppleHDA ALC283 (2 of 3)</string>
<key>Name</key>
<string>AppleHDA</string>
<key>Find</key>
<data>PYUI7BA=</data>
<key>Replace</key>
<data>PQAAAAA=</data>
</dict>
<dict>
<key>Comment</key>
<string>AppleHDA ALC283 (3 of 3)</string>
<key>Name</key>
<string>AppleHDA</string>
<key>Find</key>
<data>PYQZ1BE=</data>
<key>Replace</key>
<data>PYMC7BA=</data>
</dict>
```

Alles anzeigen

Nach

einem Neustart, sollte euer Audio in den Systemeinstellungen zu finden sein. (Kopfhörer/Mic werden erst angezeigt, wenn ihr an dessen Port was anschließt)

Soviel zu den Treibern und Komponenten. 😊

5. Feinarbeiten und DSDT Anpassung:

Für

alle, deren USB-Ports noch nicht vollständig funktionieren, empfehle ich einer DSDT Anpassung mit RehabMan's Repo im Ordner "USB" und dann für unser Notebook den Patch "usb_7-series.txt (Andere haben bei mir nicht funktioniert). Dann sollte euer USB auch lauffähig sein.

Nun zum Akku: Ihr braucht zuerst folgende Kext : <https://bitbucket.org/RehabMan...-battery-driver/downloads>

diese installiert ihr nach /S/L/E/ und müsst danach noch folgende Patche in die DSDT einfügen, da sonst nur eine mittelmäßige Akku Anzeige vorhanden ist:

Im Ordner "System" die Patche "system_Mutex.txt" und "system_OSYS.txt" ausführen, somit sollte der Akku voll und ganz angezeigt werden.

Weiteres habe ich hier nicht mehr ausgeführt an Feinarbeiten.

6. Erstellung einer SSDT zur nativen PowerManagement unterstützung:

hierzu

benötigt wir das Programm ssdtPRGen.sh-Beta (Es muss die Beta sein, da im master keine Unterstützung für unsere CPU vorhanden ist) die findet ihr hier: <https://github.com/Piker-Alpha/ssdtPRGen.sh/tree/Beta>

Ausser

diese Seite werde ich hier keine weitere Anleitung schreiben, eine Anleitung gibt es [HIER](#)

Ich

hoffe die Anleitung hat euch weiter geholfen, leider musste ich sie in zwei Teile aufteilen, wegen der Maximalen Zeichenlänge. Ihr findet zur Not (Falls die Links nicht funktionieren) alle Dateien in meiner Dropbox.

<https://www.dropbox.com/sh/j5w...IFOSHq7RzgGhdgXEa0va?dl=0>

Für Verbesserungen oder Neuerungen bin ich offen, ebenso für kritik 😊

MfG Dennis

Beitrag von „Griven“ vom 31. Mai 2015, 20:17

Ich habe dann mal beide Teile zusammengeführt 😊

Beitrag von „al6042“ vom 31. Mai 2015, 20:25

Vielen Dank für die umfangreiche Anleitung... 😊

Beitrag von „mc_killa“ vom 31. Mai 2015, 20:58

Vielen dank [Griven](#), für die Zusammenfügung, hab hier echt Panik bekommen. 😄

Gern geschehen. 😊 Ich hoffe das es einigen hilft.

Beitrag von „Griven“ vom 31. Mai 2015, 20:59

Joa, kein Problem ist ja unsere Aufgabe solche Dinge zu erledigen. Weiß aus eigener Erfahrung das es bei umfangreicheren Projekten doof ist, dass man sich nicht selbst antworten darf 😊

Beitrag von „mc_killa“ vom 31. Mai 2015, 21:16

Das muss ich allerdings auch bemängeln 😄

Wie soll man hier mit sich selbst reden, wenn das so nicht geht?! 👍

Nein kleiner Spaß, ist schon ganz gut so, um Massenspam zu vermeiden.

Beitrag von „crusadegt“ vom 31. Mai 2015, 21:17

Danke für die Anleitung 😊

Beitrag von „mc_killa“ vom 31. Mai 2015, 21:20

gern geschehen



Beitrag von „Brucklyn“ vom 31. Mai 2015, 23:15

Hi Mc_Killa

erstmal danke für diese Anleitung. Habe aber ein kleines Problem 😊

Den Stick konnte ich wunderbar erstellen sowie die config.plist ersetzen.

Nach dem Booten beim Installationsbildschirm, kommt bei mir ein Hinweis, das ich bitte eine bluetooth Tastatur anschließen soll.

Was kann ich tun ?

Beitrag von „MacGrummel“ vom 1. Juni 2015, 05:49

Ich schätze mal, dass Deine eingebaute Tastatur über PS/2 angebunden ist, was es im Mac nie gab, die meisten Notebooks aber noch immer verwenden. Und da schreibt mc_killa am Ende des ersten Teils: "**In dem nun verbliebenen Ordner "Other" löscht ihr nun alles, bis auf VoodooPS2Controller.kext und FakeSMC.kext (Da für die Zukunft nichts weiteres benötigt wird).**" Wenn Deine Tastatur jetzt nicht erkannt wird, hilft vielleicht auch eine mit USB-Kabel weiter. Den PS/2-Kext kannst Du sonst auch in die normalen Extensions packen (/S/L/E), dann sollte es aber auch mit der Notebook-Tastatur weiter gehen. Nach Bluetooth fragt Dein Rechner erst, wenn er nichts mit Kabel gefunden hat!

Beitrag von „al6042“ vom 1. Juni 2015, 07:35

Ich würde mich erstmal freuen, da der interne BT-Chip wohl direkt funktioniert.

Um aber bei deiner Geschichte weiterzukommen, solltest du prüfen, ob du BT auch im BIOS vorübergehend deaktivieren kannst, oder erstmal eine USB-Tastatur anklemmen und prüfen, ob der VoodooPS2 vorhanden und geladen ist.

Beitrag von „Brucklyn“ vom 1. Juni 2015, 11:15

Hey mit USB Maus und Tastatur hat es wunderbar geklappt.



Beitrag von „crusadegt“ vom 1. Juni 2015, 11:29

Super funktioniert denn nun alles?

Beitrag von „Brucklyn“ vom 1. Juni 2015, 15:18

Hey Joar klappt alles hat bei mir nur was länger gedauert 😊 Trotz der Super Anleitung

Beitrag von „mc_killa“ vom 1. Juni 2015, 16:01

Hi Brucklyn,

den Fehler hatte ich zwischendurch auch, bei der ersten Installation von Yosemite zumindest. Ich habe es hier nicht erwähnt, da ich es ja durch Neuinstallation von allein behoben habe (Vielleicht die USB-Schnittstellen falsch gepatcht) er hatte bei mir das mit der Bluetooth Tastatur so lange, bis ich den USB-Dongle für meine Maus einmal entfernt und wieder eingesteckt habe, danach ging die Meldung weg und Tastatur und Maus liefen einwandfrei.

Ich hab es stark auf ein USB Problem geschoben.

Ich habe mal für den Fall der Fälle drei Dateien von RehabMan in meiner Dropbox hochgeladen, sind aktuellere Versionen von VoodooPS2Controller.kext, Generic-USB3-2014-1016.kext und FakeSMC.kext.

Vielleicht helfen die, das Problem zukünftig sofort zu beheben.

Hier nochmal der Dropbox link:
<https://www.dropbox.com/sh/j5w...IFOSHq7RzgGhdgXEa0va?dl=0>

Gruß mc_killa