

Anleitung: Rebrandete Wlan-karte mit Linux nutzen

Beitrag von „Patricksworld“ vom 2. März 2017, 14:07

UPDATE: Ich habe die Schritte an einem 3 Cleaninstalls vorgenommen und es funktioniert jetzt fehlerfrei.

Der [@Sascha_77](#) bietet [hier an](#) eure Wlankarten trotz Laptopwhitelist zum laufen zu bringen.

Für Windows und Linux war diese aber bisher dann für die Tonne. Da ich aber bekannter Weise Linux liebe und meinem Laptop im Dualboot benutze (OSX / Ubuntu) möchte ich allen anderen einen Weg zeigen die Karte dennoch unter Linux zu nutzen.

Da ich selber ein PC-Noob bin bitte ich um etwas Nachsicht, wenn hier etwas technisch nicht ganz korrekt erklärt wird. Schreibt einfach einen Hinweis und ich ändere die Fehler dann nachträglich.

Überblick Schritte:

- PCI-ID Überprüfen
- Kernel von kernel.org laden
- Kernel / Kernelmodule anpassen
- Kernel kompilieren
- (Grub2 anpassen - empfohlen, aber nicht nötig)
- Kernel installieren
- Postinstall / neuladen ath9k

Vorwort:

Da zum Kompilieren der Kernel gedownloadet werden muss, sowie einige Pakete vorab aus den Paketquellen geladen werden müssen, solltet ihr eine Internetverbindung über Lan oder einen USB-Dongle herstellen.

Es werden ca 1,1 GB aus dem Internet geladen.

Spoiler anzeigen

Ich beziehe mich hier auf eine gerebrandete Atheros 9280 mit einer Intel N7260 PCI-ID und nutze Ubuntu.

Außerdem sollte jedem klar sein, das ihr nach einem Kernelupdate die Schritte erneut durchführen müsst!

Und um der VERWIRRUNG vorzubeugen. Ich benutze LINUX (UBUNTU) zum kompilieren. Es ist nur ein OSX-Theme. Also nicht das ihr auf die Idee kommt unter OSX zu kompilieren 😊

Schritt 1: PCI-ID Überprüfen

Ist die Karte getauscht könnt ihr zuerst nachsehen, was verbaut ist und welcher treiber läuft.

Das könnt ihr ganz einfach im Terminal überprüfen mit:

Code

1. lspci -nnk

```
patrick@patrick-ThinkPad-Edge-E540: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe

Kernel driver in use: lpc_ich
Kernel modules: lpc_ich
00:1f.2 SATA controller [0106]: Intel Corporation 8 Series/C220 Series Chipset Family 6-port SATA Controller 1 [AHCI mode] [8086:8c03] (rev 04)
Subsystem: Lenovo 8 Series/C220 Series Chipset Family 6-port SATA Controller 1 [AHCI mode] [17aa:5028]
Kernel driver in use: ahci
Kernel modules: ahci
00:1f.3 SMBus [0c05]: Intel Corporation 8 Series/C220 Series Chipset Family SMBus Controller [8086:8c22] (rev 04)
Subsystem: Lenovo 8 Series/C220 Series Chipset Family SMBus Controller [17aa:5028]
Kernel modules: i2c_i801
02:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller [10ec:8168] (rev 10)
Subsystem: Lenovo RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller [17aa:5028]
Kernel driver in use: r8169
Kernel modules: r8169
03:00.0 Network controller [0280]: Intel Corporation Wireless 7260 [8086:08b2] (rev 01)
Subsystem: Intel Corporation Wireless-N 7260 [8086:4262]
Kernel modules: iwlwifi
patrick@patrick-ThinkPad-Edge-E540:~$ █
```

Man erkennt dass das Kernelmodul (Treiber) iwlwifi geladen wird. Das wäre grundsätzlich ja auch korrekt für die entsprechende PCI-ID. Allerdings ist es unser Ziel jetzt den ath9k treiber zu laden, da dieser für die AR9280 zuständig ist. Dafür müssen wir den Kernel anpassen und neu kompilieren.

Schritt 2: Kernel downloaden

Öffnet ein Terminal und wechselt in euer Homeverzeichnis (eigentlich der standartordner)

Code

1. cd ~

legt euch einen neuen ordner an und wechselt in das Verzeichnis:

Code

1. mkdir kernelneu
2. cd kernelneu

den entsprechenden Kernel downloaden und entpacken (und umbenennen):

Code

1. wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.10.1.tar.xz
2. tar xf linux-4.10.1.tar.xz
3. mv linux-4.10.1 linux-4.10.1-custom

Anschließend in das Verzeichnis wechseln.

Code

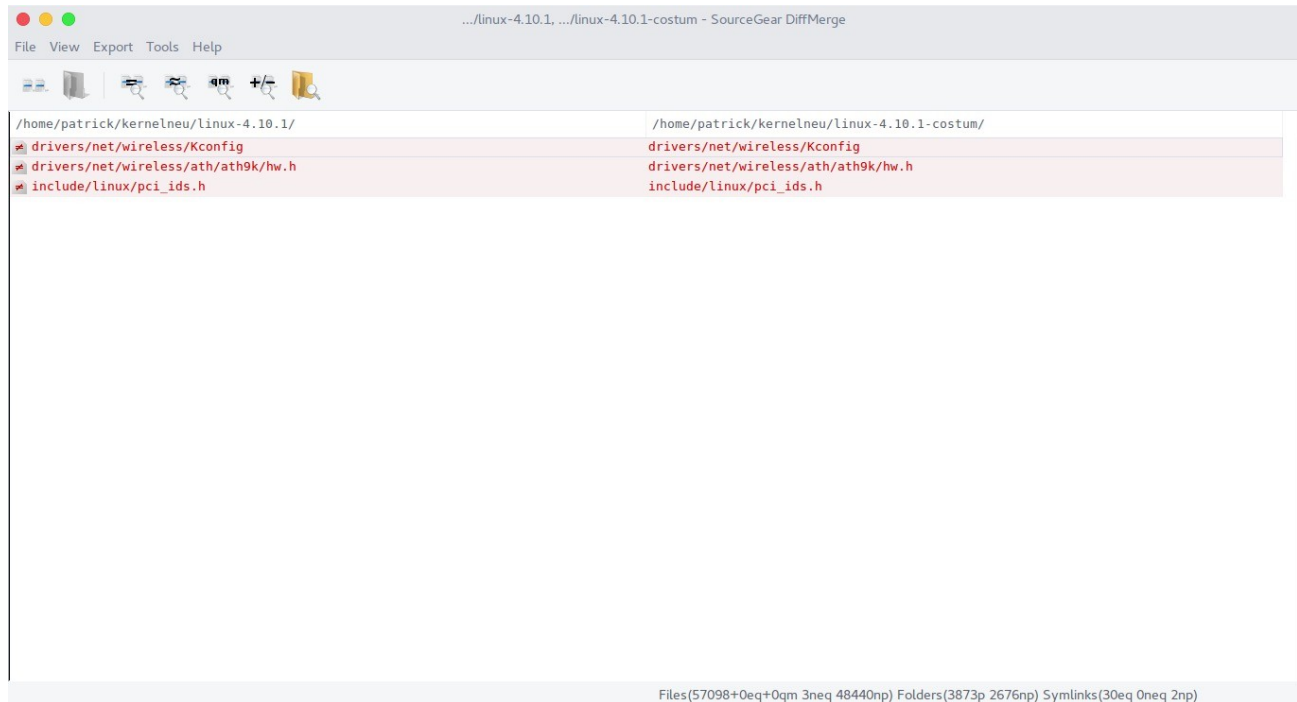
1. cd linux-4.10.1-custom/

Schritt 3: Kernel / Kernelmodule anpassen

Ich habe lediglich 3 Dateien angepasst. Eine muss mit Sicherheit angepasst

werden. Eine ist fakultativ und die 3 bin ich unsicher. Müsste man nocheinmal ohne testen.

Folgende 3 Dateien habe ich angepasst:



Wir müssen mindestens im ath9k die pci-id anpassen. Unsicher bin ich mir mit den pci_ids.h Eintrag. Der stört aber auf jeden Fall nicht. Ich glaube aber es dürfte auch ohne gehen.

Also starten wir mit dem ath9k treiber.

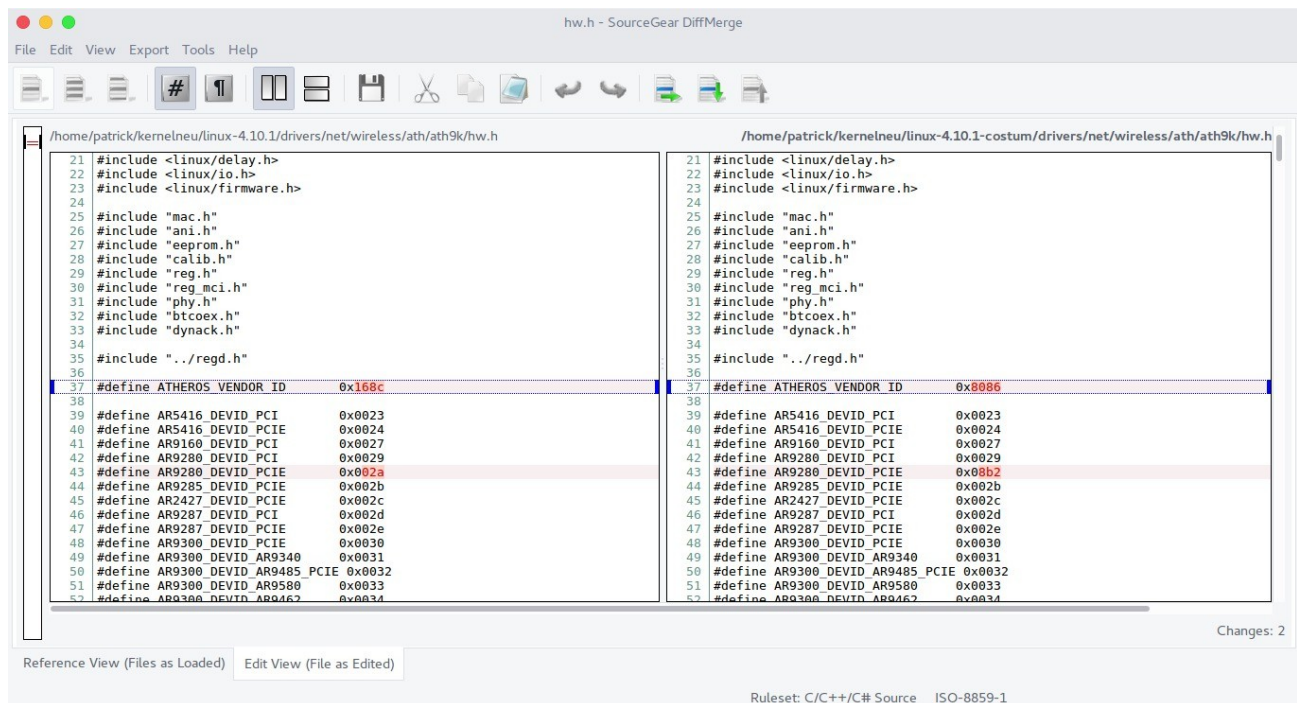
Die entscheidende Information befindet sich in der hw.h Datei. Also öffnen wir diese mit unserem Texteditors des Vertrauens (bei mir gedit)

Code

1. gedit drivers/net/wireless/ath/ath9k/hw.h

Dort tauscht ihr die entsprechenden PCI-ID's mit den vorher ermittelten und speichert das ganze ab.

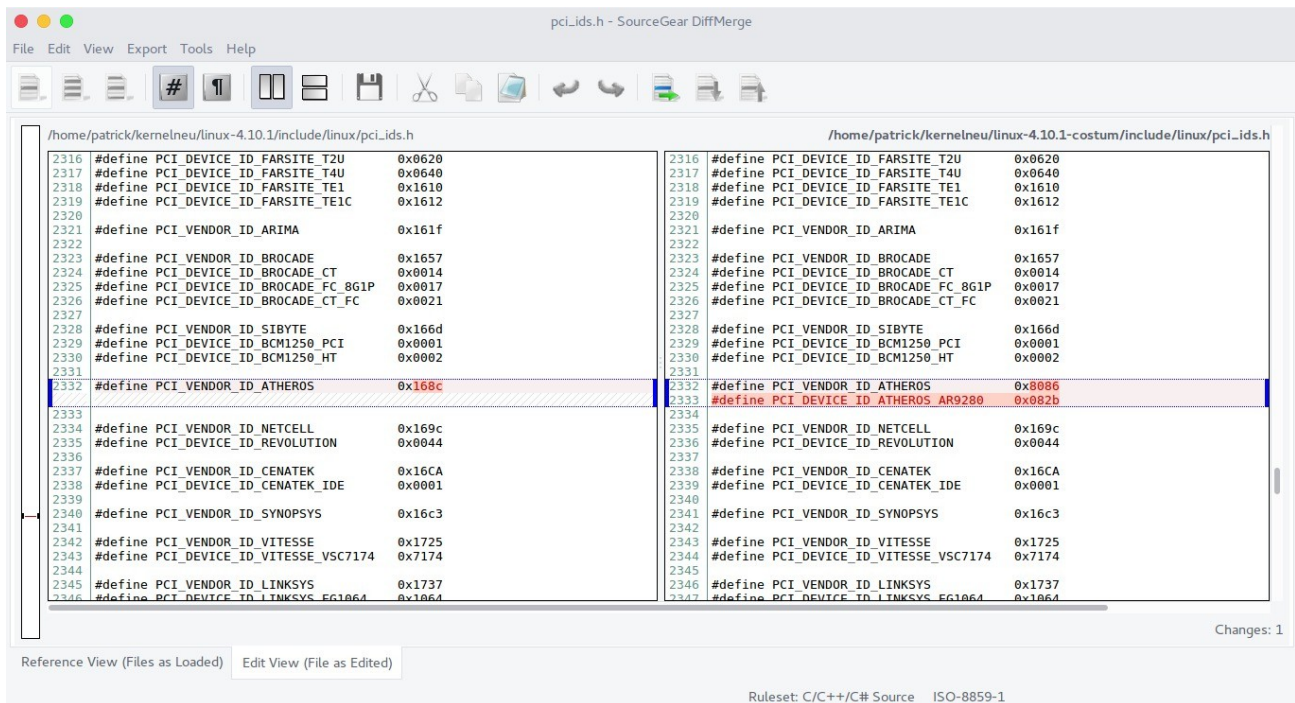
links original / rechts verändert



Als nächstes passen wir die pci_ids.h an:
(ich bin mir nicht sicher ob das zwangsläufig sein muss)

Code

1. gedit include/linux/pci_ids.h

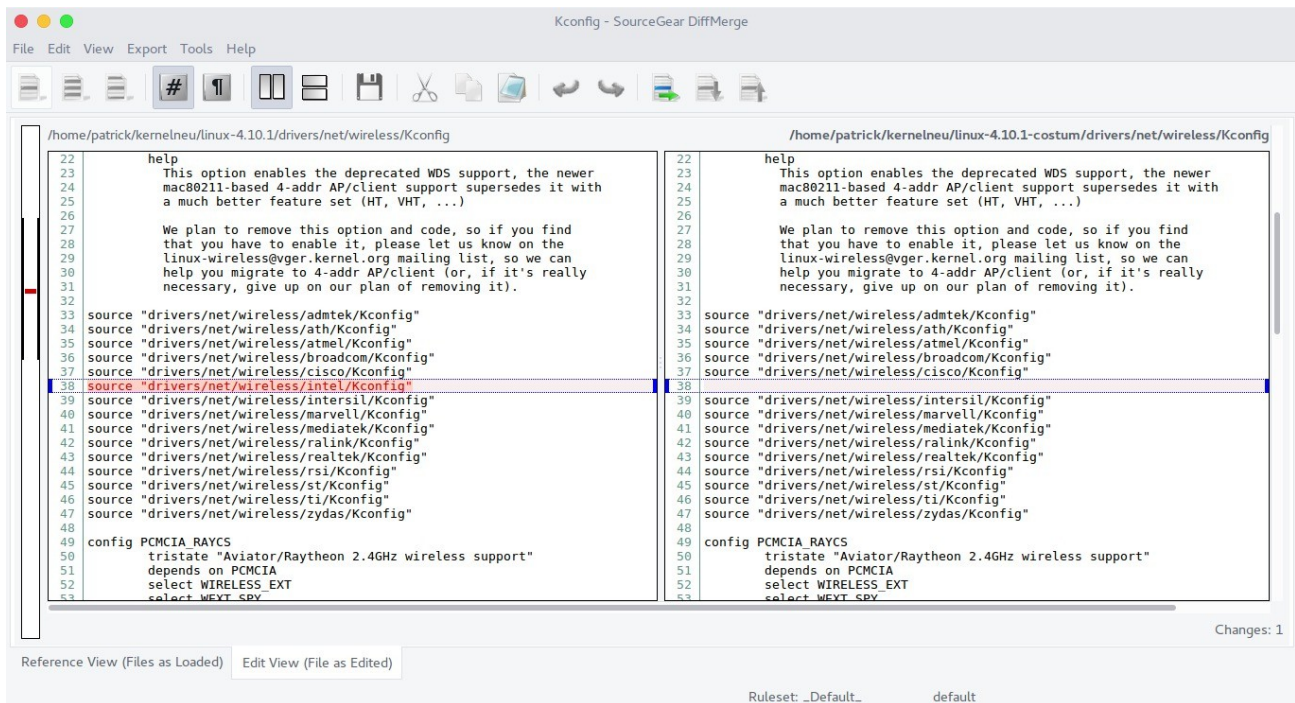


Und zu guter letzt passen wir noch die Kconfig an.

Das hat nur den Hintergrund das ich den ilwifi treiber garnicht mitkompilieren will. Somit kann das auch nicht fälschlicherweise geladen werden. Man könnte das Modul genauso gut nachträglich blockieren. Doch um mir den Ärger zu ersparen entferne ich einfach die entsprechende Zeile.

Code

1. gedit drivers/net/wireless/Kconfig



Und das waren auch schon die Anpassungen.

Der Rest ist Linux Standard Prozedur.

Schritt 4: Kernel Kompilieren

Generell benötigt ihr zum kompilieren folgende Pakete / Programme:

- **linux-source**
- **build-essential**
- **kernel-package**
- **libssl-dev**

Ihr könnt die einfach mit folgendem Befehl im Terminal laden und installieren:

Code

1. `sudo apt-get install linux-source build-essential kernel-package libssl-dev`

Zum Kompilieren müsst ihr folgende Befehle nacheinander ins Terminal reinschmeißen:

Code

1. `cp /boot/config-`uname -r` .config`
2. `yes "" | make oldconfig`
3. `make -j8 bzImage modules`

Schritt 5: Grub2 - anpassen (fakultativ)

So weit so gut. Vor dem Kernel installieren solltet ihr jedoch sicherstellen das ihr zur not den "alten" Kernel laden könnt. Damit ihr auch ein Auswahlmenü bekommt um im Notfall den Kernel manuell auswählen zu können, benutze ich immer [Grub-Customizer](#) um wenigstens 10 sekunden zur Auswahl zu haben.

Installation im Terminal:

Fremdpaketquelle freischalten:

Code

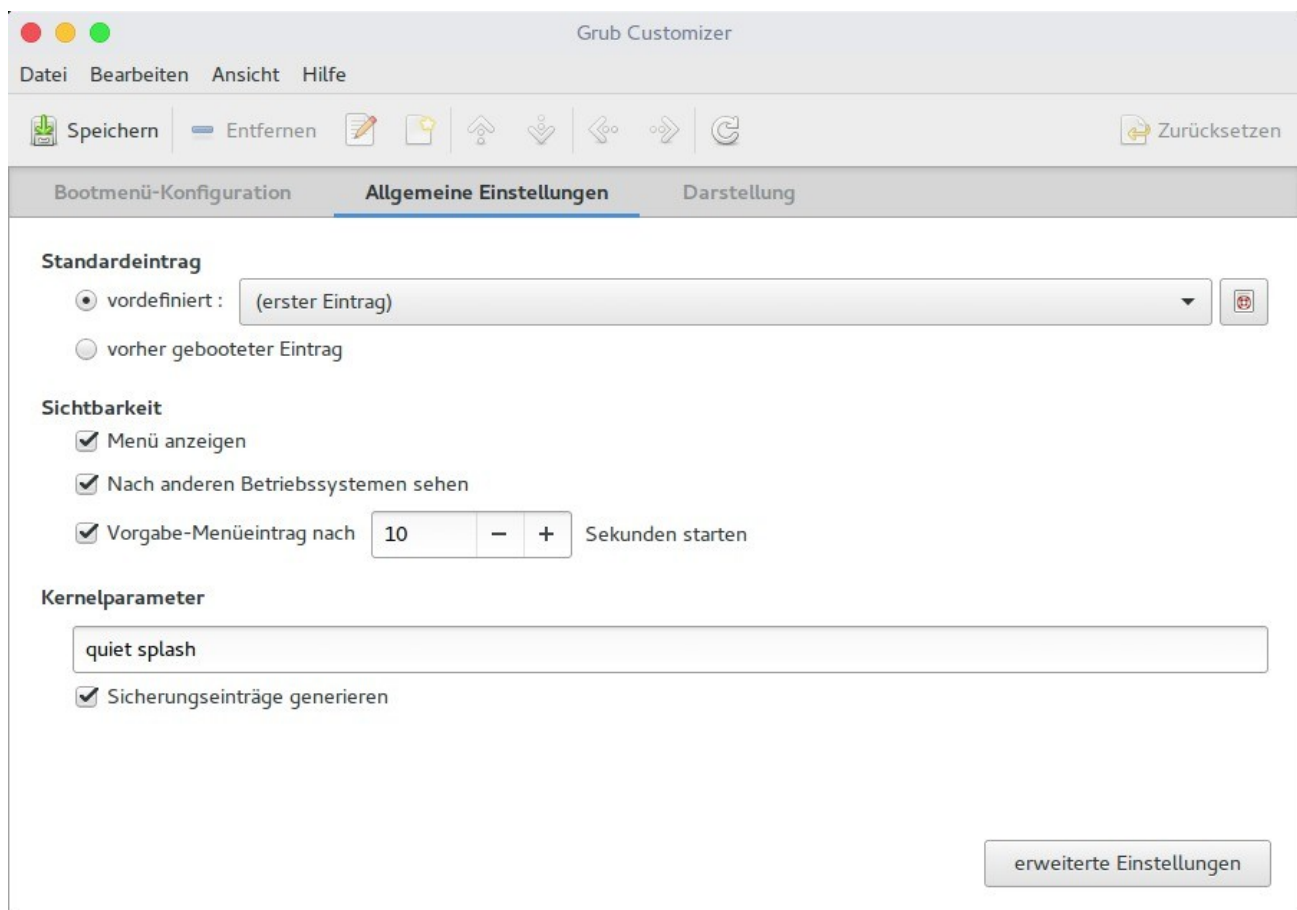
1. `sudo add-apt-repository ppa:danielrichter2007/grub-customizer`

Und installieren:

Code

1. `sudo apt-get update`
2. `sudo apt-get install grub-customizer`

Anschließend folgende Einstellungen übernehmen:



Das ist wie gesagt nur für Angsthassen 😊

Schritt 6: Kernel installieren

fast geschafft:

Code

1. `sudo make modules_install install`

--> Neustart

Schritt 7: Postinstall / neuladen ath9k

ath9k konfigurieren/ Device anlegen:

Code

1. `echo 'install ath9k /sbin/modprobe --ignore-install ath9k; /bin/echo "8086 08b2" > /sys/bus/pci/drivers/ath9k/new_id' | sudo tee /etc/modprobe.d/ath9k_options.conf`

den alten ath9k entfernen und neu laden

Code

1. `sudo modprobe -rfv ath9k`
2. `sudo modprobe -v ath9k`

Dannach sollte es schon funktionieren.

Damit der ath9k bei systemstart automatisch geladen wird führen wir noch folgenden code aus:

Code

1. `echo ath9k | sudo tee -a /etc/modules`

Überprüfen könnt ihr das mit einem erneuten:

Code

1. `lspci -nnk`

```
patrick@patrick-ThinkPad-Edge-E540: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
:8c16] (rev d4)  
Kernel driver in use: pcieport  
Kernel modules: shpchp  
00:1c.4 PCI bridge [0604]: Intel Corporation 8 Series/C220 Series Chipset Family PCI Express Root Port #5 [8086  
:8c18] (rev d4)  
Kernel driver in use: pcieport  
Kernel modules: shpchp  
00:1f.0 ISA bridge [0601]: Intel Corporation HM87 Express LPC Controller [8086:8c4b] (rev 04)  
Subsystem: Lenovo HM87 Express LPC Controller [17aa:5028]  
Kernel driver in use: lpc_ich  
Kernel modules: lpc_ich  
00:1f.2 SATA controller [0106]: Intel Corporation 8 Series/C220 Series Chipset Family 6-port SATA Controller 1  
[AHCI mode] [8086:8c03] (rev 04)  
Subsystem: Lenovo 8 Series/C220 Series Chipset Family 6-port SATA Controller 1 [AHCI mode] [17aa:5028]  
Kernel driver in use: ahci  
Kernel modules: ahci  
00:1f.3 SMBus [0c05]: Intel Corporation 8 Series/C220 Series Chipset Family SMBus Controller [8086:8c22] (rev 0  
4)  
Subsystem: Lenovo 8 Series/C220 Series Chipset Family SMBus Controller [17aa:5028]  
Kernel modules: i2c_i801  
02:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ether  
net Controller [10ec:8168] (rev 10)  
Subsystem: Lenovo RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller [17aa:5028]  
Kernel driver in use: r8169  
Kernel modules: r8169  
03:00.0 Network controller [0280]: Intel Corporation Wireless 7260 [8086:08b2] (rev 01)  
Subsystem: Intel Corporation Wireless-N 7260 [8086:4262]  
Kernel driver in use: ath9k  
patrick@patrick-ThinkPad-Edge-E540:~$ █
```

Ich freue mich auf Rückmeldungen.

MFG Patrick