

# Hardware unter Linux

It's all the manufacturer's fault?

Christian Perle

3. November 2007

# Gliederung

- 1 Übersicht
- 2 Voraussetzungen für Treiber
- 3 Kategorien von Treibern
- 4 Probleme und deren Ursachen
- 5 Praktisches Vorgehen

# Was brauchen Entwickler, um einen Treiber zu schreiben?

# Was brauchen Entwickler, um einen Treiber zu schreiben?

- Erfahrung in Linux-Kernelprogrammierung

# Was brauchen Entwickler, um einen Treiber zu schreiben?

- Erfahrung in Linux-Kernelprogrammierung
- Kenntnisse darüber, wie die Hardware anzusteuern ist

# Was brauchen Entwickler, um einen Treiber zu schreiben?

- Erfahrung in Linux-Kernelprogrammierung
- Kenntnisse darüber, wie die Hardware anzusteuern ist
  - Durch Informationen des Chipsatz-Herstellers

# Was brauchen Entwickler, um einen Treiber zu schreiben?

- Erfahrung in Linux-Kernelprogrammierung
- Kenntnisse darüber, wie die Hardware anzusteuern ist
  - Durch Informationen des Chipsatz-Herstellers
  - Durch Reverse Engineering

## Optimal: Treiber im offiziellen Kernel-Tree

- Hat einen gewissen Reifegrad erreicht

## Optimal: Treiber im offiziellen Kernel-Tree

- Hat einen gewissen Reifegrad erreicht
- Wird aktiv gepflegt

## Optimal: Treiber im offiziellen Kernel-Tree

- Hat einen gewissen Reifegrad erreicht
- Wird aktiv gepflegt
- Steht unter GPL

## Optimal: Treiber im offiziellen Kernel-Tree

- Hat einen gewissen Reifegrad erreicht
- Wird aktiv gepflegt
- Steht unter GPL
- Beispiele

## Optimal: Treiber im offiziellen Kernel-Tree

- Hat einen gewissen Reifegrad erreicht
- Wird aktiv gepflegt
- Steht unter GPL
- Beispiele
  - Intel ipw2100, ipw2200

## Optimal: Treiber im offiziellen Kernel-Tree

- Hat einen gewissen Reifegrad erreicht
- Wird aktiv gepflegt
- Steht unter GPL
- Beispiele
  - Intel ipw2100, ipw2200
  - ZyDAS zd1211rw

## Optimal: Treiber im offiziellen Kernel-Tree

- Hat einen gewissen Reifegrad erreicht
- Wird aktiv gepflegt
- Steht unter GPL
- Beispiele
  - Intel `ipw2100`, `ipw2200`
  - ZyDAS `zd1211rw`
  - RaLink `rt2xxx` (noch nicht integriert)

## Gut: Treiber out of Tree, vollständig Open Source

- Steht unter GPL oder ist GPL-kompatibel

## Gut: Treiber out of Tree, vollständig Open Source

- Steht unter GPL oder ist GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, lässt sich durch offenen Quelltext aber anpassen

## Gut: Treiber out of Tree, vollständig Open Source

- Steht unter GPL oder ist GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, lässt sich durch offenen Quelltext aber anpassen
- Beispiele

## Gut: Treiber out of Tree, vollständig Open Source

- Steht unter GPL oder ist GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, lässt sich durch offenen Quelltext aber anpassen
- Beispiele
  - RaLink `rt2570` (legacy, serialmonkey project)

## Gut: Treiber out of Tree, vollständig Open Source

- Steht unter GPL oder ist GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, lässt sich durch offenen Quelltext aber anpassen
- Beispiele
  - RaLink `rt2570` (legacy, serialmonkey project)
  - RaLink `rt73` (legacy, serialmonkey project)

## Gut: Treiber out of Tree, vollständig Open Source

- Steht unter GPL oder ist GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, lässt sich durch offenen Quelltext aber anpassen
- Beispiele
  - RaLink `rt2570` (legacy, serialmonkey project)
  - RaLink `rt73` (legacy, serialmonkey project)
  - RealTek `rtl818x` (vendor driver)

## Gut: Treiber out of Tree, vollständig Open Source

- Steht unter GPL oder ist GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, lässt sich durch offenen Quelltext aber anpassen
- Beispiele
  - RaLink `rt2570` (legacy, serialmonkey project)
  - RaLink `rt73` (legacy, serialmonkey project)
  - RealTek `rtl818x` (vendor driver)
  - ZyDAS `zd1211` (vendor driver)

# Erträglich bis schlecht: Treiber out of Tree, teilweise Closed Source

- Nicht GPL-kompatibel

## Erträglich bis schlecht: Treiber out of Tree, teilweise Closed Source

- Nicht GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, selten anpaßbar

## Erträglich bis schlecht: Treiber out of Tree, teilweise Closed Source

- Nicht GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, selten anpaßbar
- Wenn der Autor/Hersteller entscheidet, den Treiber nicht mehr weiterzuentwickeln, ist der Treiber tot (kann von niemand anderem weitergepflegt werden)

## Erträglich bis schlecht: Treiber out of Tree, teilweise Closed Source

- Nicht GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, selten anpaßbar
- Wenn der Autor/Hersteller entscheidet, den Treiber nicht mehr weiterzuentwickeln, ist der Treiber tot (kann von niemand anderem weitergepflegt werden)
- Beispiele

## Erträglich bis schlecht: Treiber out of Tree, teilweise Closed Source

- Nicht GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, selten anpaßbar
- Wenn der Autor/Hersteller entscheidet, den Treiber nicht mehr weiterzuentwickeln, ist der Treiber tot (kann von niemand anderem weitergepflegt werden)
- Beispiele
  - Fritz fcdsl (Kernel 2.4 only)

## Erträglich bis schlecht: Treiber out of Tree, teilweise Closed Source

- Nicht GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, selten anpaßbar
- Wenn der Autor/Hersteller entscheidet, den Treiber nicht mehr weiterzuentwickeln, ist der Treiber tot (kann von niemand anderem weitergepflegt werden)
- Beispiele
  - Fritz fcdsl (Kernel 2.4 only)
  - Fast alle Treiber für Winmodems

## Erträglich bis schlecht: Treiber out of Tree, teilweise Closed Source

- Nicht GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, selten anpaßbar
- Wenn der Autor/Hersteller entscheidet, den Treiber nicht mehr weiterzuentwickeln, ist der Treiber tot (kann von niemand anderem weitergepflegt werden)
- Beispiele
  - Fritz `fcdsl` (Kernel 2.4 only)
  - Fast alle Treiber für Winmodems
  - Atheros `madwifi` (Hardware Abstraction Layer ist Closed Source), wird langfristig durch `ath5k` abgelöst

## Erträglich bis schlecht: Treiber out of Tree, teilweise Closed Source

- Nicht GPL-kompatibel
- Nicht immer zum aktuellen Kernel-Tree kompatibel, selten anpaßbar
- Wenn der Autor/Hersteller entscheidet, den Treiber nicht mehr weiterzuentwickeln, ist der Treiber tot (kann von niemand anderem weitergepflegt werden)
- Beispiele
  - Fritz fcdsl (Kernel 2.4 only)
  - Fast alle Treiber für Winmodems
  - Atheros `madwifi` (Hardware Abstraction Layer ist Closed Source), wird langfristig durch `ath5k` abgelöst
  - Intel `ipw3945` (benötigt Closed Source Daemon)

## Erträglich bis schlecht: Closed Source Xorg-Treiber

- Laufen nicht im Kernel, aber immerhin mit root-Rechten

## Erträglich bis schlecht: Closed Source Xorg-Treiber

- Laufen nicht im Kernel, aber immerhin mit root-Rechten
- Hängen von teilw. Closed Source Kernelmodulen ab

## Erträglich bis schlecht: Closed Source Xorg-Treiber

- Laufen nicht im Kernel, aber immerhin mit root-Rechten
- Hängen von teilw. Closed Source Kernelmodulen ab
- Beispiele

## Erträglich bis schlecht: Closed Source Xorg-Treiber

- Laufen nicht im Kernel, aber immerhin mit root-Rechten
- Hängen von teilw. Closed Source Kernelmodulen ab
- Beispiele
  - ATI `fglrx`  
Bessert sich in Zukunft, AMD/ATI hat Spezifikationen für einige Radeon-Chipsätze offengelegt

## Erträglich bis schlecht: Closed Source Xorg-Treiber

- Laufen nicht im Kernel, aber immerhin mit root-Rechten
- Hängen von teilw. Closed Source Kernelmodulen ab
- Beispiele
  - ATI `fglrx`  
Bessert sich in Zukunft, AMD/ATI hat Spezifikationen für einige Radeon-Chipsätze offengelegt
  - NVidia `nvidia`

## Erträglich bis schlecht: Closed Source Xorg-Treiber

- Laufen nicht im Kernel, aber immerhin mit root-Rechten
- Hängen von teilw. Closed Source Kernelmodulen ab
- Beispiele
  - ATI `fglrx`  
Bessert sich in Zukunft, AMD/ATI hat Spezifikationen für einige Radeon-Chipsätze offengelegt
  - NVidia `nvidia`
  - Matrox `mga/hal`

## Spezialität bei WLAN-Karten: ndiswrapper

- Lädt Windows-Treiber nach

## Spezialität bei WLAN-Karten: ndiswrapper

- Lädt Windows-Treiber nach
- Ist effektiv ein Closed Source Treiber

## Spezialität bei WLAN-Karten: ndiswrapper

- Lädt Windows-Treiber nach
- Ist effektiv ein Closed Source Treiber

### Achtung

Persönliche Meinung: **Nicht** benutzen, da Hersteller sonst eine Entschuldigung haben, keine Linux-Treiber zu schreiben bzw. keine Informationen für Treiberentwickler bereitzustellen.

## Informationspolitik der Hersteller und die Folgen

- Hardware-Ansteuerung durch Referenzimplementation oder Dokumentation offengelegt

## Informationenpolitik der Hersteller und die Folgen

- Hardware-Ansteuerung durch Referenzimplementation oder Dokumentation offengelegt
  - Beste Ausgangsbasis

## Informationspolitik der Hersteller und die Folgen

- Hardware-Ansteuerung durch Referenzimplementation oder Dokumentation offengelegt
  - Beste Ausgangsbasis
- Informationen nur unter NDA

# Informationspolitik der Hersteller und die Folgen

- Hardware-Ansteuerung durch Referenzimplementation oder Dokumentation offengelegt
  - Beste Ausgangsbasis
- Informationen nur unter NDA
  - Weniger Code Review und kein verteiltes Entwickeln möglich

# Informationspolitik der Hersteller und die Folgen

- Hardware-Ansteuerung durch Referenzimplementation oder Dokumentation offengelegt
  - Beste Ausgangsbasis
- Informationen nur unter NDA
  - Weniger Code Review und kein verteiltes Entwickeln möglich
- Überhaupt keine Informationen

# Informationspolitik der Hersteller und die Folgen

- Hardware-Ansteuerung durch Referenzimplementation oder Dokumentation offengelegt
  - Beste Ausgangsbasis
- Informationen nur unter NDA
  - Weniger Code Review und kein verteiltes Entwickeln möglich
- Überhaupt keine Informationen
  - reverse engineering

## Informationspolitik der Hersteller und die Folgen

- Hardware-Ansteuerung durch Referenzimplementation oder Dokumentation offengelegt
  - Beste Ausgangsbasis
- Informationen nur unter NDA
  - Weniger Code Review und kein verteiltes Entwickeln möglich
- Überhaupt keine Informationen
  - reverse engineering
  - Treiber potentiell instabiler, weil das Hardwareverhalten beobachtet/erraten werden musste

# Informationspolitik der Hersteller und die Folgen

- Lizenzprobleme, Patente

# Informationspolitik der Hersteller und die Folgen

- Lizenzprobleme, Patente
  - Besonders bei „dummer“ Hardware, viele Funktionen/Verfahren werden im Treiber realisiert

## Informationspolitik der Hersteller und die Folgen

- Lizenzprobleme, Patente
  - Besonders bei „dummer“ Hardware, viele Funktionen/Verfahren werden im Treiber realisiert
- Gesetzliche Bestimmungen (Frequenzregulierung von WLAN-Karten)

# Informationspolitik der Hersteller und die Folgen

- Lizenzprobleme, Patente
  - Besonders bei „dummer“ Hardware, viele Funktionen/Verfahren werden im Treiber realisiert
- Gesetzliche Bestimmungen (Frequenzregulierung von WLAN-Karten)
  - Besonders bei „dummer“ Hardware, Einschränkungen werden im Treiber realisiert (siehe `madwifi`, `ipw3945`)

# Hausgemachte Probleme bei WLAN

- Für lange Zeit fehlte ein standardisiertes Konfigurationsinterface

# Hausgemachte Probleme bei WLAN

- Für lange Zeit fehlte ein standardisiertes Konfigurationsinterface
- Viele Treiber mit eigenen Konfigurationstools

# Hausgemachte Probleme bei WLAN

- Für lange Zeit fehlte ein standardisiertes Konfigurationsinterface
- Viele Treiber mit eigenen Konfigurationstools
- Konsolidierung: Jährliche WLAN-Summits, neuer WLAN-Stack seit Kernel 2.6.22 (`mac80211`)

# Hausgemachte Probleme bei WLAN

- Für lange Zeit fehlte ein standardisiertes Konfigurationsinterface
- Viele Treiber mit eigenen Konfigurationstools
- Konsolidierung: Jährliche WLAN-Summits, neuer WLAN-Stack seit Kernel 2.6.22 (`mac80211`)
- WLAN-Treiber werden schrittweise auf `mac80211` umgestellt

## Karte XYZ-42, WTF?

- Gleicher Kartenname bedeutet **nicht** gleicher Chipsatz!

## Karte XYZ-42, WTF?

- Gleicher Kartenname bedeutet **nicht** gleicher Chipsatz!
  - Chipsätze werden oft gewechselt, um die Karte ein paar Cent billiger anbieten zu können

## Karte XYZ-42, WTF?

- Gleicher Kartename bedeutet **nicht** gleicher Chipsatz!
  - Chipsätze werden oft gewechselt, um die Karte ein paar Cent billiger anbieten zu können
  - Solange Geiz geil ist, wird das so bleiben. . .

## Karte XYZ-42, WTF?

- Gleicher Kartename bedeutet **nicht** gleicher Chipsatz!
  - Chipsätze werden oft gewechselt, um die Karte ein paar Cent billiger anbieten zu können
  - Solange Geiz geil ist, wird das so bleiben...
- Identifikation der Hardware durch `lspci -v -nn` oder `lsusb`

## Karte XYZ-42, WTF?

- Gleicher Kartename bedeutet **nicht** gleicher Chipsatz!
  - Chipsätze werden oft gewechselt, um die Karte ein paar Cent billiger anbieten zu können
  - Solange Geiz geil ist, wird das so bleiben...
- Identifikation der Hardware durch `lspci -v -nn` oder `lsusb`
- Im Netz nach der Vendor/Product-ID suchen

## Karte XYZ-42, WTF?

- Gleicher Kartename bedeutet **nicht** gleicher Chipsatz!
  - Chipsätze werden oft gewechselt, um die Karte ein paar Cent billiger anbieten zu können
  - Solange Geiz geil ist, wird das so bleiben...
- Identifikation der Hardware durch `lspci -v -nn` oder `lsusb`
- Im Netz nach der Vendor/Product-ID suchen
- Windows-Treiber herunterladen, \*.inf-Datei untersuchen

## D-Link DWL-G122, aber welche Revision?



# D-Link DWL-G122, Rev.C1



lsusb

```
Bus 004 Device 012: ID 07d1:3c03 D-Link System
```

## D-Link DWL-G122, Rev.C1, Treiberinstallation

- Chipsatz RaLink RT2573, Treiber `rt73` nicht im Kernel

## D-Link DWL-G122, Rev.C1, Treiberinstallation

- Chipsatz RaLink RT2573, Treiber `rt73` nicht im Kernel
- Build-Tools installieren

```
apt-get install build-essential
```

```
apt-get install linux-headers-`uname -r`
```

## D-Link DWL-G122, Rev.C1, Treiberinstallation

- Chipsatz RaLink RT2573, Treiber `rt73` nicht im Kernel
- Build-Tools installieren

```
apt-get install build-essential  
apt-get install linux-headers-`uname -r`
```

- Treiber herunterladen und übersetzen

```
wget http://rt2x00.serialmonkey.com/rt73-cvs-daily.tar.gz  
tar xzf rt73-cvs-daily.tar.gz  
cd rt73-cvs-2*/Module  
make
```

## D-Link DWL-G122, Rev.C1, Treiberinstallation

- Chipsatz RaLink RT2573, Treiber `rt73` nicht im Kernel
- Build-Tools installieren

```
apt-get install build-essential  
apt-get install linux-headers-`uname -r`
```

- Treiber herunterladen und übersetzen

```
wget http://rt2x00.serialmonkey.com/rt73-cvs-daily.tar.gz  
tar xzf rt73-cvs-daily.tar.gz  
cd rt73-cvs-2*/Module  
make
```

- Kein Betrieb mit `wpa_supplicant` möglich, WPA nur über `iwpriv` konfigurierbar

# US RoboticsUSR5423



lsusb

```
Bus 004 Device 014: ID 0baf:0121 U.S. Robotics
```

## US RoboticsUSR5423, Treiberinstallation

- Chipsatz ZyDAS zd1211b, Treiber `zd1211rw` im Kernel seit Version 2.6.18

# US Robotics USB5423, Treiberinstallation

- Chipsatz ZyDAS zd1211b, Treiber `zd1211rw` im Kernel seit Version 2.6.18
- Firmware auf  
<http://linuxwireless.org/en/users/Drivers/zd1211rw>  
(offiziell von ZyDAS freigegeben)

## US RoboticsUSR5423, Treiberinstallation

- Chipsatz ZyDAS zd1211b, Treiber `zd1211rw` im Kernel seit Version 2.6.18
- Firmware auf  
<http://linuxwireless.org/en/users/Drivers/zd1211rw>  
(offiziell von ZyDAS freigegeben)
- Funktioniert mit `wpa_supplicant`

Bis jetzt leider sehr selten zu sehen. . .



Linux-Support auf der Packung angegeben (Chipsatz ZyDAS zd1211b).

## Identifikation über Windows-Treiber

- Karte: Linksys WPC300N (Cardbus)

## Identifikation über Windows-Treiber

- Karte: Linksys WPC300N (Cardbus)
- Windows-Treiber auf <http://www-de.linksys.com/>

## Identifikation über Windows-Treiber

- Karte: Linksys WPC300N (Cardbus)
- Windows-Treiber auf <http://www-de.linksys.com/>
- Archiv entpacken:  
`unzip WPC300N_EU_1.1.00.019.zip`

## Identifikation über Windows-Treiber

- Karte: Linksys WPC300N (Cardbus)
- Windows-Treiber auf <http://www-de.linksys.com/>
- Archiv entpacken:  
unzip WPC300N\_EU\_1.1.00.019.zip
- Datei `drivers/XP/net5416.inf` untersuchen:

```
[...]  
%Linksys_WPC300N% = ... PCI\VEN_168C&DEV_0023...  
%Linksys_WMP300N% = ... PCI\VEN_168C&DEV_0023...  
[...]
```

## Identifikation über Windows-Treiber

- PCI-ID 168c:0023, Chipsatz Atheros AR5008X

## Identifikation über Windows-Treiber

- PCI-ID 168c:0023, Chipsatz Atheros AR5008X
- `modinfo ath_pci | grep -i '168c.*0023'`  
liefert nichts

## Identifikation über Windows-Treiber

- PCI-ID 168c:0023, Chipsatz Atheros AR5008X
- `modinfo ath_pci | grep -i '168c.*0023'`  
liefert nichts
- Karte ist noch zu neu für den `madwifi`-Treiber

## Identifikation über Windows-Treiber

- PCI-ID 168c:0023, Chipsatz Atheros AR5008X
- `modinfo ath_pci | grep -i '168c.*0023'`  
liefert nichts
- Karte ist noch zu neu für den `madwifi`-Treiber
- nicht kaufen oder kaufen und hoffen, dass der Treiber  
entsprechend erweitert wird

# That's all, folks!

Fragen?