

ENGLISH TRANSLATION**Contents**

- 1 Projektbeschreibung
- 2 Was ist DD-WRT?
- 3 Unterstützte Geräte
- 4 Komplette Featureliste
- 5 Die verschiedenen Pakete
- 6 Unterschiede zwischen den 6 verschiedenen Dateien
 - ◆ 6.1 Betrieb als Access-Point
 - ◆ 6.2 Betrieb als Client
- 7 Richtig Flashen
 - ◆ 7.1 Der Hard-Reset oder 30/30/30-Reset
 - ◇ 7.1.1 Hinweis Asus-Geräte
 - ◇ 7.1.2 Hinweis WRT54
 - ◇ 7.1.3 Umgang mit Backupkonfigurationen
 - ◆ 7.2 Grundlagen
 - ◆ 7.3 Vorgehensweise
 - ◆ 7.4 Besonderheiten beim Flashen von...
 - ◇ 7.4.1 WRT54G v2.2
 - ◇ 7.4.2 WRT54G/S v4.0 WRT54GL
 - ◇ 7.4.3 WRT54GS
 - ◇ 7.4.4 WRT54G v5 - v6
 - ◇ 7.4.5 Buffalo WHR-G54S, WHR-HP-G54 und WHR-G125
 - ◇ 7.4.6 Buffalo WHR-HP-G54 - ändern der Boardflags
 - ◆ 7.5 Wiederbeleben
- 8 Das Web-Interface
 - ◆ 8.1 Beschreibung der einzelnen Einstellungen
- 9 Konfigurationsbeispiele
 - ◆ 9.1 WRT54G im Client-Modus
 - ◆ 9.2 WRT54GL als Repeater Bridge
 - ◇ 9.2.1 Verschlüsselung des Virtuellen Interfaces
 - ◆ 9.3 WRT an einer FritzBox (o.a.) für ein Nachbarschaftsnetz
 - ◆ 9.4 DD-WRT Router als WDS einrichten um ein Netz mit mehreren Routern aufzubauen, an denen sich Clients anmelden können
 - ◇ 9.4.1 Vorbereitungen
 - ◇ 9.4.2 Einstellungen in DD-WRT
 - 9.4.2.1 I. Folgende Daten sind im Vorfeld zu notieren
 - 9.4.2.2 II. Für alle Router (Den einen Internet- und die Vielzahl von WDS-Routern) gleich
 - 9.4.2.3 III. Für den Internet-Router
 - 9.4.2.4 IV. Für alle anderen (WDS) Router
 - 9.4.2.5 V. Testen
- 10 Linux inside
 - ◆ 10.1 Wichtige Linuxbefehle der Shell
 - ◆ 10.2 Traffic control (traffic shaping) mit tc
 - ◆ 10.3 selber compilieren
 - ◆ 10.4 Skript-Schnipsel
 - ◆ 10.5 Antennen ausrichten

- ◊ [10.5.1 Client-Modus](#)
 - ◊ [10.5.2 AP-Modus](#)
 - ◆ [10.6 Software installieren mit ipkg](#)
 - ◆ [10.7 Port-Knocking](#)
 - ◊ [10.7.1 Was ist Port-Knocking?](#)
 - ◊ [10.7.2 Wozu braucht man Port-Knocking?](#)
 - ◊ [10.7.3 Voraussetzungen](#)
 - ◊ [10.7.4 Installation der benötigten Pakete](#)
 - ◊ [10.7.5 Startskripte erstellen](#)
 - [10.7.5.1 Erstellen eines Ordners für Skripte](#)
 - [10.7.5.2 Erstellen eines allgemeinen Startup-Skripts](#)
 - [10.7.5.3 Erstellen eines Skriptes für den Start von knockd](#)
 - [10.7.5.4 Automatischen Start einrichten](#)
 - ◊ [10.7.6 knockd.conf anpassen](#)
 - ◊ [10.7.7 Anwendungsbeispiele](#)
 - ◊ [10.7.8 Die Client-Tools](#)
 - ◊ [10.7.9 Quellen](#)
 - [11 Bootvorgang](#)
 - [12 Hardware](#)
 - ◆ [12.1 Modifikationen](#)
 - ◊ [12.1.1 Dual Serial Port](#)
 - ◊ [12.1.2 SD-Card Reader](#)
 - ◊ [12.1.3 JTAG-Adapter](#)
 - ◊ [12.1.4 WRT350N Serial Port](#)
 - ◊ [12.1.5 WRT350N-Antennen Mod](#)
 - ◆ [12.2 Version herausfinden](#)
 - ◆ [12.3 Informationen über Hardware](#)
 - ◆ [12.4 NVRAM](#)
 - ◆ [12.5 Overclocking der CPU](#)
 - [13 JFFS2](#)
 - [14 Erweiterungen](#)
 - ◆ [14.1 USB Support](#)
 - ◆ [14.2 Installation eines Samba Servers](#)
 - ◆ [14.3 FTP-Server](#)
 - [15 RFlow Collector](#)
 - [16 Trouble Shooting](#)
 - [17 Ich möchte mitmachen](#)
 - [18 Alphabetischer Index](#)
-

Achtung: Diese Wiki wurde schon lange nicht mehr aktualisiert. Viele Artikel behandeln DD-WRT Versionen die nicht mehr aktuell sind (DD-WRT Version 22pre5) und Probleme die so nicht mehr existieren.

Die Community hat wahrscheinlich kein Interesse mehr an einer Deutschen-Wiki.

Attention: This Wiki is out of date!

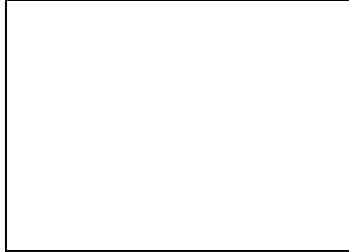
it seems that the german community is not longer interested in this german-wiki.

Projektbeschreibung

Dieses Projekt beschäftigt sich mit der Dokumentation von DD-WRT, einer freien Firmwaremodifikation, die hauptsächlich für den WRT54G(GS) von Linksys gedacht ist. Wie jedes Media-Wiki lebt auch dieses vom Mitmachen. Jeder darf, und soll Artikel verfassen, editieren und diskutieren.

Viel Spaß!

Was ist DD-WRT?



Das Interface der aktuellen DD-WRT v23/v24 Versionen

DD-WRT ist eine freie Firmware für die Router Linksys WRT54G und Linksys WRT54GS. In der Version V23SP1 vom 16.05.2006 werden auch das Gerät Asus WL-500 Deluxe und verschiedene Router von Allnet, Buffalo, Belkin, Motorola, Siemens (Gigaset SE505), Ravo und Askey unterstützt. Mit der Version V24 werden neue Geräte, auf dem 802.11N-Draft Standard basierend, unterstützt. Siehe dazu auch die offizielle Hardware-Support-Liste.

DD-WRT basiert auf einem minimalen Linux und Code, der von Linksys unter der Software Lizenz GPL veröffentlicht wurde.

Zunächst erweiterte die Firma Sveasoft die veröffentlichten Quelltexte um neue Funktionen. Als Sveasoft jedoch 20 Dollar für den Zugriff auf die Supportforen ihrer Firmware und den Downloadbereich verlangte, gründeten die Dresdner Sebastian Gottschall ("BrainSlayer", BS) und Matthias Engel ("mateng") auf Basis dieser Software das **DD-WRT**-Projekt. Seither erschienen weiterentwickelte Versionen der Firmware auf der Homepage von **DD-WRT**. Mit der Version 23 schließlich orientierte sich Brainslayer stärker an openwrt, einer modular angelegten Firmware, die bis vor kurzem kein Webinterface hatte und bei Geschwindigkeit, Stabilität und Funktionalität Vorteile gegenüber Sveasoft als Firmwarebasis versprach. Zwischen den beiden Projekten werden nach Brainslayers Angaben Ideen ausgetauscht, so dass beide von einander profitieren.

DD-WRT bietet einen deutlich größeren Funktionsumfang als die Original Firmware von Linksys. Es sind auch einige versteckte Optionen freigeschaltet. Desweiteren kommen noch einige Zusatzfunktionen hinzu, wie etwa das Einbinden von Samba-Freigaben, WDS, RADIUS-Authentifizierung.

Unterstützte Geräte

Modell	Version	Prozessor	Flash	RAM	USB
Avila Gateworks GW2347	-	XSCALE IXP 425 266 MHz	8 MB	32 MB	keine

DD-WRT_Doku_(DE)

Avila Gateworks GW2348-2	-	XSCALE IXP 425 266 MHz	8 MB	32 MB	keine
Avila Gateworks GW2348-4	-	XSCALE IXP 425 533 MHz	16 MB	64 MB	keine
Linksys WRT54G	2.0, 2.2, 3.0, 3.1	Broadcom 4712 200 MHz	4 MB	16 MB	keine
Linksys WRT54G	4.0	Broadcom 5352 200 MHz	4 MB	16 MB	keine
Linksys WRT54GL	1.0, 1.1	Broadcom 5352 200 MHz	4 MB	16 MB	keine
Linksys WRT54GS	1.0	Broadcom 5352 200 MHz	8 MB	32 MB	keine
Linksys WRT54GS	1.1	Broadcom 4712 200 MHz	8 MB	32 MB	keine
Linksys WRT54GS	4.0	Broadcom 5352 200 MHz	4 MB	16 MB	keine
LaFonera		Atheros AR5315 183 MHz	4 MB	16 MB	keine
ASUS WL-500g	???	Broadcom 4710 125 MHz	4 MB	16 MB	1x v1.1
ASUS WL-500g Deluxe	ab dem 2005-11-01 Release von v23beta2	Broadcom 5365 200 MHz	4 MB	32 MB	2x v2.0
ASUS WL-500g Premium		Broadcom 4704 266 MHz	8 MB	32 MB	2x v2.0
Siemens SE 505	V1	Broadcom 4710 125 MHz	4 MB	16 MB	keine
Siemens SE 505	V2	Broadcom 4712 200 MHz	4 MB	8 MB	keine (USB-Mod)
Siemens SX550/SX550i	-	Broadcom 4710 125 MHz	4 MB	16 MB	keine
Motorola WR850G	V1	Broadcom 4710 125 MHz	4 MB	16 MB	keine
Motorola WR850G	V2	Broadcom 4712 200 MHz	4 MB	16 MB	keine
Motorola WR850G	V3	Broadcom 4712 200 MHz	4 MB	16 MB	keine
Buffalo WBR2G54S		Broadcom 4712 200 MHz	4 MB	16 MB	keine
Buffalo WHR-G54S		Broadcom 5352 200 Mhz	4 MB	16 MB	keine
Buffalo WHR-HP-G54		Broadcom 5352 200 Mhz	4 MB	16 MB	keine

- Linksys WRT54G v1.0, v1.1, v2.0, v2.2, v3.0, v3.1, v4.0, v5.0, v5.1, v6.0 (ab v5.0 nur Micro)
 - ◆ **NICHT** v7.0
- Linksys WRT54GL v1.0, v1.1 (ab dem 2005-11-24-Release von v23beta2)
- Linksys WRT54GS v1.0, v1.1, v2.0, v2.1, v3.0, v4.0, v5.0, v5.1, v6.0 (ab v5.0 nur Micro)
- Asus WL-500G Deluxe (ab dem 2005-11-01 Release von v23beta2)
- Asus WL-500g Premium
- Siemens SE505 ([Link zum Forum](#))
- Motorola WR850G (ab dem 2005-11-07 Release von v23beta2)
- Buffalo WBR2G54S
- Buffalo WHR-G54S
- Buffalo WHR-HP-G54

Hardware-Support-Liste

Komplette Featureliste

- ◆ Hotspot-Portal (Sputnik Agent, Chillispot)
- ◆ PPTP-VPN-Server
- ◆ 2-Wege-Bandbreiten-Management (inkl. P2P, VoIP, IM)
- ◆ SSH-Client und -Server (Dropbear)
- ◆ Telnet
- ◆ Startup-, Firewall-, und Shutdown-Skripte
- ◆ WDS-Repeater-Modus
- ◆ Client-Modus (unterstützt mehrere angebundene Geräte)
- ◆ Ad-Hoc-Modus
- ◆ OSPF-Routing
- ◆ RIP2-Routing
- ◆ Power-Boost-Funktion (bis 251 mW Sendeleistung)
- ◆ Antennenauswahl
- ◆ Statische DHCP-Adresszuweisung
- ◆ Zusätzliche DDNS-Unterstützung
- ◆ Wireless MAC-Adressen duplizieren
- ◆ VLAN-Unterstützung (WAN-Port als normalen LAN-Port)
- ◆ WPA über WDS
- ◆ WPA/TKIP mit AES
- ◆ WPA2-Unterstützung
- ◆ Client-Modus WPA
- ◆ Client-Isolation-Modus
- ◆ P2P Blocking & Bandbreiten-Management (Gnutella, Kazaa, etc.)
- ◆ Port Triggering
- ◆ Port Forwarding (30 Einträge möglich)
- ◆ Wake-on-Lan
- ◆ Remote Syslog
- ◆ Remote Ntop-Statistik
- ◆ XBOX-Kaid-Unterstützung
- ◆ SNMP
- ◆ IPv6-Unterstützung
- ◆ Sicheres Backup und Restore
- ◆ Reset bei Firmware-Upgrade
- ◆ Status-Anzeige für Wireless Clients und WDS mit Systembetriebszeit, Prozessorlast u. a.
- ◆ Site Survey
- ◆ Remote NTP-Server
- ◆ Webserver
- ◆ QoS
- ◆ FTP-Server
- ◆ 12 verschiedene Sprachen
- ◆ Unterstützt auch die neuen WRT54G V3, V3.1, V4 und WRT54GS V2.1, V3, V4 Modelle
- ◆ und vieles mehr ...

Die verschiedenen Pakete

(Achtung dieser Artikel ist nicht mehr aktuell)

Aktuelle Version 23 SP1 vom 16.05.2006 ([Changelog](#)) wurden die Pakete folgendermaßen aufgeteilt:

- [dd-wrt.v23_sp1_micro](#) - beinhaltet nicht: Chillisport, NoCat, RFlow, Kaid, Samba Client, SNMP, IPv6, MMC/SD Card Support, SSH, PPTP/PPTP Client, UPnP, HTTPS Support für Web Management, OpenVPN, SIPatH. Die Datei ist kleiner als 2MB. Sie ist bestimmt für Router mit weniger als 2MB Flash Speicher. (z.B. Linksys WRT54G version 5). Jeder Router **sollte** in der Lage sein, diese Firmwareversion verwenden, so auch die Routerversionen vor WRT54G v5. für Geräte die wenig Speicher und Flash Platz haben. Es wird jetzt auch WRT54G/GS v5 unterstützt. Für weitergehene Informationen zum Flashen von WRT54G/GS v5 Geräten bitte die [Vxworkskiller](#) Seite besuchen.
- [dd-wrt.v23_sp1_mini](#) - beinhaltet nicht: chillisport, nocat, rflow, kaid, samba client, SNMP, IPv6, MMC/SD Card Support, und HTTPS support für web management. Es beinhaltet fast alles was man braucht, lässt aber noch genug flash Platz für Anpassungen mit jffs.
- [dd-wrt.v23_sp1_std](#) - beinhaltet nicht: OpenVPN, SIPatH. Es beinhaltet fast alles was man für die professionelle Nutzung braucht inklusive snmp, hotspot features wie chillisport und sputnik (tm).
- [dd-wrt.v23_sp1_voip](#) - beinhaltet nicht: Kaid, MMC/SD Card Supp., OpenVPN - beinhaltet zusätzlich SIP Gateway Routing system für managing SIP capable Geräte in einem privaten Netzwerk.
- [dd-wrt.v23_sp1_vpn](#) - beinhaltet nicht: Kaid, RADVD, SIPatH. - beinhaltet einen voll featured openvpn client für Netzwerke mit hohen Sicherheitsanforderungen.

	Micro	Mini	Standard	VOIP	VPN
Chillisport			X	X	X
HTTPS Support for Web Management			X	X	X
IPv6			X	X	X
kaid			X		
MMC/SD Card Support			X		X
NoCat			X	X	X
OpenVPN					X
PPTP/PPTP Client		X	X	X	X
RADVD	X	X	X	X	
RFlow			X	X	X

DD-WRT_Doku_(DE)

Samba Client		X	X	X
SIPatH			X	
SSH	X	X	X	X
UPnP	X	X	X	X

Unterschiede zwischen den 6 verschiedenen Dateien

Dateiname	Beschreibung
dd-wrt.v23_<type>_asus.trx	Web interface version zum flashen. Siehe auch <u>Flash Your Asus WL-500G Deluxe</u>
dd-wrt.v23_<type>_generic.bin	Generic version zum flashen über das web interface auf allen unterstützten Geräten (inclusive Linksys WRT54G/GL/GS) und zum flashen von <u>Siemens SE505</u> mit boot wait tftp on 192.168.2.1
dd-wrt.v23_<type>_wrt54g.bin	tftp versions für WRT54G. You CAN use this to flash via web interface but do so only AFTER you have done so using the mini version. These versions were just made specifically for TFTPing to those routers. (v5/v6 note: Since the WRT54G/GS v5-v6 uses a modified WAP54Gv3 once made 'linux ready', it will not accept these standard WRT54G/GS firmwares. You must use the 'generic' build for TFTPing to these units).
dd-wrt.v23_<type>_wrt54gs.bin	tftp versions for WRT54GS. You CAN use this to flash via web interface but do so only AFTER you have done so using the mini version. These versions were just made specifically for TFTPing to those routers.
dd-wrt.v23_<type>_wrt54gsv4.bin	tftp versions for WRTGSv4. You CAN use this to flash via web interface but do so only AFTER you have done so using the mini version. These versions were just made specifically for TFTPing to those routers.
dd-wrt.v23_<type>_wrtsl54gs.bin	tftp versions for WRTSL54GS. You CAN use this to flash via web interface but do so only AFTER you have done so using the mini version. These versions were just made specifically for TFTPing to those routers.
dd-wrt.v23_<type>_moto.trx	For initial <u>Flash Your Motorola WR850G (Micro and Mini builds ONLY)</u>

Betrieb als Access-Point

Als erstes sollte ich einen Computer haben. Als nächstes sollte schon ein Internetanschluss vorhanden sein. Dieser sollte am Besten als DSL-Flatrate vorhanden sein. Wenn diese Voraussetzungen gegeben sind, sollte ich mir einen **Linksys WRT54G/GS/L** der richtigen Hardwareversion (s. u.) anschaffen. Diesen bekomme ich im gut sortierten IT-Fachhandel.

Betrieb als Client

Bei Nutzung des WRT54G im Client-Modus brauche ich neben einer Alternativ-Firmware wie DD-WRT oder Alchemy, die den Client-Modus unterstützt, einen Internetprovider, der in meiner Nähe (möglichst mit Sichtkontakt!) einen Access Point (AP) betreibt und dessen Zugangsdaten (Login bzw. Freischaltung meiner MAC-Adressen am AP). Außerdem empfiehlt es sich die 2-dB-Stabantennen gegen leistungsstärkere Antennen, z. B. BiQuad-Antennen, zu tauschen und diese Antennen außerhalb des Hauses anzubringen. Leistungsstärker heißt: Antennen mit einem höheren dB/dBi-Wert, also statt mit nur 2 dB, besser mit 12 dB. Das wird durch eine stärkere Richtwirkung der Abstrahlung erreicht: also statt kugel- oder diskusförmiger Abstrahlung (Beispiel: Glühbirne) bei der Stabantenne, eine mehr keulenförmige Abstrahlung (Beispiel: Taschenlampe). Die Antenne muß zum WRT54G einen RP-TNC-Stecker haben (Gewinde innen mit Kelch). Diese werden bei Ebay bereits ab 1,99 Euro das Stück angeboten.

Drahtlos angebundene Computer sind am WRT54G im Client-Modus leider nicht zu betreiben - man braucht dann einen zweiten Access Point bzw. muß die PCs per LAN-Kabel an den vier LAN-Ports anbinden!

- ◆ Anleitung WLAN-Client mit DD-WRT
- ◆ Anleitung Client <-> AP mit 2 WRT54G(S) (auf englisch)

Richtig Flashen

Nach dem Einspielen einer neuen Firmware arbeitet die neue Firmware erst einmal mit den Einstellwerten, die in der vorhergehenden Firmware eingestellt waren. Ungünstige Einstellungen können dazu führen, dass das Gerät bereits beim Booten der neuen Firmware abstürzt und auf keinerlei Eingaben reagiert.

Von einer Firmwareversion, die als Up- bzw. Downgrade für eine bestimmte Firmware angeboten wird, darf man minimal erwarten, dass das nicht passiert, wenn die Einstellwerte der Ausgangsfirmware vor dem Flashen auf die Standardwerte zurückgesetzt werden. Allerdings darf man nicht erwarten, dass die neue Firmware mit den Standard-Einstellwerten der alten Firmware fehlerfrei arbeitet. Es wird lediglich sichergestellt, dass die neue Firmware soweit funktioniert, dass sich die Standardwerte für die neue Firmware setzen lassen.

Der Hard-Reset oder 30/30/30-Reset

Um Fehlern beim Flashen vorzubeugen, sollte die alte Firmware vor dem Flashen und die neue Firmware nach dem Flashen auf ihre Standardwerte gesetzt werden. Dazu muss der Hard- oder auch 30/30/30-Reset vor und nach jedem Firmware-Upgrade / Downgrade durchgeführt werden! Er bewirkt das Löschen der Einstellungen im NVRAM und setzt die DD-WRT-Firmware auf Standard-Werte zurück.

1. Das Gerät ist eingeschaltet (am Strom angestöpselt), nun drückt und hält man die Reset-Taste auf der Rückseite für 30 Sekunden.
2. Ohne Loslassen der Reset-Taste, trennt man das Gerät vom Strom und hält die Reset-Taste für weitere 30 Sekunden.
3. Ohne Loslassen der Reset-Taste, das Gerät wieder in Betrieb nehmen und erst nach weiteren 30 Sekunden die Reset-Taste loslassen.

Hinweis Asus-Geräte

Bei Asus-Geräten ist zu beachten, dass der normale 30/30/30-Reset das Gerät nach dem 3. Schritt in den Recovery-Mode wechselt. Daher muss der Router nach den 30 Sekunden des 3. Schrittes nochmals vom Strom getrennt werden und nach ein paar Sekunden wieder angeschlossen werden damit er normal startet.

Hinweis WRT54

Beim WRT54 kann es passieren, dass nach dem 3. Schritt die Power-Leuchte durchgängig blinkt. Einfach nochmals für 5 Sekunden vom Strom trennen und wieder anstöpseln. Dann sollte der Router normal starten und leuchten.

Umgang mit Backupkonfigurationen

Aus den oben genannten Gründen sollten sie eine Backupkonfigurationen nur in die gleiche Firmwareversion zurückspielen, mit der die Backupkonfiguration erstellt wurde. **Verwenden Sie keine gespeicherten Backupkonfigurationen, wenn Sie ein anderes Firmware-Build nutzen (verschiedene svn Build-Nummern).**

Grundlagen

Wer es noch nicht weiß: Die Linksys-Router können eine andere Firmware aufgespielt bekommen, die weitere ungeahnte und zum Teil vielleicht auch unnötige Einstellungsmöglichkeiten bieten. Jedoch erweist sich dieses Aufspielen einer anderen Firmware öfter als fehlerhaft. Den derzeitigen Fehlerberichten zufolge funktioniert dann der Router vorerst nicht mehr, jedoch gibt es in so einem Fall auch Lösungen, 2, 3.

Doch um diesem "falschen" Flashen vorzubeugen, diene diese Anleitung. *(da dieses Wiki aber noch im Aufbau ist, so sei beachtet, dass diese Art und Weise "richtig" zu flashen nur EINE Möglichkeit von vielen ist. Wer sich also strikt an diese Anleitung hält und trotzdem Probleme bekommt, kann diesen Beitrag selbst ändern, dem Editor dieses Beitrages eine Rüge geben oder sonstwas gemäß der Vernunft tun)*
Übrigens sind die Links zum Teil wie "<http://routerip/Management.asp>" notiert, für "routerip" sollte hier die eigene Router-IP eingegeben werden. In der Regel 192.168.1.1; bzw. nach Schritt 4 sowieso.

Vorgehensweise

1. Sich zum einen die gewünschte DD-WRT-Firmware herunterladen und entpacken, so dass die *.bin-Datei im Ordner des PCs liegt. Diese gibts zur Zeit hier: [DD-WRT](#)

1b. DD-WRT Version 22pre5 gibt es in drei Versionen: generic, basic und full. "Generic" ist, wie der Name schon sagt, eine generische Version auch für andere Router, die mit ähnlichen Chipsätzen ausgestattet sind und auf denen DD-WRT auch läuft (z. B. Asus WL500g Deluxe). "Basic" ist die normale Standardversion und "full" eine erweiterte Version die zusätzlich Dropbear, Chillispot, Samba und kaid enthält. Wer die nicht braucht, ist mit der Basic-Version gut bedient. Zusätzlich sind noch jeweils für den WRT54g oder WRT54gs angepasste Basic- und Full-Versionen erhältlich. [Infos](#). Desweiteren sei angemerkt, dass Prefinal-Versionen teilweise noch kleinere Bugs (Funktionsfehler) aufweisen können, daher werden sie immer vorerst zu Testzwecken herausgegeben. Gleichzeitig enthalten sie aber auch die meisten Verbesserungen und viele alte behobene Fehler. Unversierte Benutzer sollten also nach persönlichen Bedürfnissen abwägen, welche Version für sie in Frage kommt.

2. Als nächstes sicher gehen, dass kein anderer PC per WLAN oder auch LAN auf den Router zugreift, außer der, welcher die Firmware aufspielen soll. Dieser "Flash-PC" sollte per LAN - also verkabelt - an den Router angeschlossen sein! *Es soll zwar Benutzer geben, die per WLAN flashen und es klappt, aber von dieser Methode wird dringlichst abgeraten!* Also alle Kabel - auch am WAN-Port! - außer dem des Stromanschlusses abziehen und nach Möglichkeit alle WLAN-User informieren, sie sollen mal ihr WLAN abschalten und auch keine Verbindungsversuche starten (die Verbindung wird ja sowieso gleich gekappt (in Schritt 3))

3. Jetzt kann, wenn erwünscht, noch von den Einstellungen ein Backup gemacht werden und zwar unter [Administration/Backup](#) und dort unter "[HERE \(gilt nur für DD-WRT-Versionen\)](#)" die Datei auf dem Rechner gespeichert werden. **Aber:** Auch wenn es bei manchen funktioniert: Es wird davon abgeraten, dieses Einstellungen-Backup für die neue Firmware zu benutzen! Es sollte alles neu eingestellt werden! *Das Backup dient also nur dazu bei der Wiederherstellung der noch in diesem Schritt genutzten Firmware, sich das zum Teil mühselige Einstellen zu ersparen.*

3b. Wer schon eine andere "3rd Party Firmware" wie etwa DD-WRT installiert hat, sollte noch "boot-wait" einschalten. Bei DD-WRT-Firmware unter: [administration/management](#)

4. Nun muss ein "Factory-Reset" durchgeführt werden. Bei manchen geht es auch ohne, aber wir wollen ja "richtig" flashen. Der Reset ist ganz einfach: An der Rückseite des Routers den "Resetknopf" mit einem spitzen Werkzeug (ein Bleistift tut auch, aber aufpassen, dass nichts von der Mine abbricht und im Gerät bleibt, besser einen Kugelschreiber nehmen) für etwa 30 Sekunden lang gedrückt halten und danach loslassen; dass etwas passiert, sieht man am Blinken der LEDs an der Vorderseite des Routers. Infolgedessen auf der [Setup](#)-Hauptseite prüfen, ob auch alle Einstellungen auf Default-Werten sind, also die eigenen Einstellungen verworfen wurden. Andernfalls den Reset-Vorgang nochmals durchführen. **!!!** Sollte nach Loslassen die DMZ-LED konstant 5 Sekunden an - 5 Sekunden aus blinken, so muss der Router einmal kurz vom Stromnetz genommen und wieder angesteckt werden. Fängt sich der Router immer noch nicht, dann nochmals die Reset-Taste, aber nur kurz, drücken. Wenn auch danach die DMZ-LED blinkt, nochmals einen "Factory Reset" probieren. Andernfalls stimmt da etwas nicht und das [Forum](#) hilft da bestimmt weiter. (Solche Fälle sind mir aber nicht bekannt)

5. Jetzt kann die Firmware geflasht werden auf der [Administration/Firmware Upgrade](#)-Seite. Die *.bin-Datei von 1. auswählen, auf den Upgrade-Button klicken und warten - richtig lange warten mitunter. Wer abbricht und den Stromstecker zieht, ist selbst schuld, denn dann kann es sein, dass das Gerät nicht ohne Mühe wieder

zum Laufen gebracht werden kann. DD-WRT Versionen haben mitunter noch eine Option, vor dem Flashen auszuwählen, ob "Reset after flashing to:" ausgeführt werden soll. Hier empfiehlt es sich, auf "Firmware Defaults" zu stellen.

5b. Sollte wie unter 4. die DMZ-LED blinken, so auch wie in 4. beschrieben vorgehen. Das Upgrade sollte komplett sein, wenn entweder eine Bestätigung "upgrade complete" oder ähnliches erscheint, oder die Power- und die Lan-Port-LED leuchten und euch nichts komisch vorkommt. Ein kurzes Trennen vom Stromnetz kann mitunter schon kleinere Problemchen lösen ...

6. Jetzt können alle Einstellungen vorgenommen werden. User/Passw. für die Weboberfläche des Routers ist: **root/admin**.

Wenn es nach dem Flashen Probleme gab, so hilft derzeit im Forum auf Englisch eventuell [dieser Beitrag](#). Eine [Anleitung](#) hier im Wiki ist im Aufbau.

Besonderheiten beim Flashen von...

WRT54G v2.2

Es scheint beim Flashen via TFTP nötig zu sein, die vergebene ip-adresse zu missachten und tftp auf die 192.168.1.1 zu setzen:

```
$ ifconfig eth1
eth1      Protokoll:Ethernet  Hardware Adresse 00:04:61:4C:45:05
          inet Adresse:192.168.1.20  Bcast:192.168.1.255  Maske:255.255.255.0
...
```

Bei Erfolg sieht der ganze Vorgang dann so aus:

```
$ tftp
tftp> mode binary
tftp> status
Not connected.
Mode: octet Verbose: off Tracing: off
Rexmt-interval: 5 seconds, Max-timeout: 25 seconds
tftp> put dd-wrt.v23_mini_wrt54g.bin
No target machine specified.
tftp> connect 192.168.1.1
tftp> put dd-wrt.v23_mini_wrt54g.bin
Sent 2417664 bytes in 4.3 seconds
tftp>
```

(ab dem put-befehl dürfte es um die 5sec dauern) Der Put-Befehl führte bei einigen Geräten auch schon vor dem Erlöschen der vier LAN_LEDs zum Erfolg der sich in kurzzeitigem sehr schnellen Blinken der LED #1 zeigte.

WRT54G/S v4.0 WRT54GL

Nur die DD-WRT Versionen v23 und höher unterstützen diese Router Version. Desweiteren kann aufgrund einer Limitierung der Original Linksys Firmware (lässt nur ein Update mit einer Firmware bis zu 3MByte zu) für dieses Gerät zunächst nur eine v23 mini Version aufgespielt werden (siehe generelle Anleitung oben). Sobald diese jedoch erfolgreich eingespielt wurde besteht diese Begrenzung nicht mehr und man kann beliebig zu anderen Firmware Versionen wie v23 basic oder voip (nicht v2X oder v1X sonst ist der Linksys v4.0 kaputt) wechseln.

WRT54GS

ACHTUNG bei TFTP-Uploads: WRT54 "GS" - Modelle können NUR am Port 1 per TFTP geflasht werden. Bei einigen Hardware-Revisionen ist der Port 1 am Switch der 4. Slot.

WRT54G v5 - v6

Sehen hierzu [Vxworkskiller.de](http://www.vxworkskiller.de)

Buffalo WHR-G54S, WHR-HP-G54 und WHR-G125

Bitte für die Buffalo Router WHR-G54S und WHR-HP-G54 nur die DD-WRT v23 SP1 und höher verwenden. Frühere Versionen unterstützen diese beiden Geräte nicht. Der Buffalo Router WHR-G125 wird ab DD-WRT v24 unterstützt .

Die Standard-IP-Adresse der Buffalo-Router lautet 192.168.11.1 mit Benutzernamen root ohne Passwort. Nach dem Flashen lautet die Standard-IP-Adresse 192.168.1.1 mit Benutzernamen root und Passwort admin. Dies gilt es am Ende des Flashens zu beachten. Die heruntergeladene Binärdatei der DD-WRT Firmware befindet sich im Pfad "LW:\Dokumente und Einstellungen\%Benutzername%", wobei LW meist das Laufwerk C: ist und Benutzername stellvertretend für den aktuellen Benutzer ist.

Anmerkungen vor dem Flashen

- Haben Sie einmal auf die DD-WRT Firmware gewechselt, können Sie die normale Firmware von Buffalo nicht mehr einspielen. Weitere Informationen findet man unter <http://forum.bsr-clan.de/viewtopic.php?t=8184&highlight=whrg54s+back+firmware>.
- Die von der Buffalo-Firmware verwendete LED in Form eine Brücke, wenn der WHR-G54S oder der WHR-HP-G54 als Bridge arbeitet - der Schiebeschalter auf der Unterseite - steht mit der DD-WRT Firmware nicht mehr zur Verfügung und ist nur während des Bootprozesses kurz in Betrieb.
- Es ist nicht nötig, die zweistufige Flashprozedur ähnlich des WRT54G durchzuführen. Die Buffalo Router akzeptieren das komplette Image in einem Rutsch.
- Innerhalb der DD-WRT Firmware kann die Weboberfläche zum Upgraden/Downgraden zwischen den DD-WRT Versionen verwendet werden. Vorsicht: unterhalb der DD-WRT Version v23 SP1 ist der Support für die Buffalo Router noch nicht enthalten.

Flashen auf die DD-WRT Firmware

1. Um unnötigen Problemen aus dem Weg zu gehen, sollte man vor dem Flashen den Router in die Werkseinstellungen zurücksetzen. Auf der Unterseite befindet sich der INIT-Knopf. Dieser muss bei angeschaltetem Router gedrückt gehalten werden, bis auf der Frontblende die rote LED erscheint. In Ausnahmefällen muss er bis zu 30 Sekunden gedrückt gehalten werden, bis der komplette Reset-to-Default durchgelaufen ist.
2. Den Stromstecker am Router abziehen.
3. Da der Router beim Starten auf der IP-Adresse 192.168.11.1 steht, muss derjenige PC, von dem aus geflasht werden soll, eine statische IP-Adresse aus dem 192.168.11.0/24 Netzwerk haben. Dies kann die IP-Adresse 192.168.11.2 mit Subnetzmaske 255.255.255.0 sein. Für Windows XP sollte dieser Screenshot (in englischer Sprache) weiterhelfen. Zuerst über Start -> Einstellungen -> Systemsteuerung die Netzwerkverbindungen öffnen. Dort mit der rechten Maustaste die LAN-Verbindung auswählen und im Kontextmenü "Eigenschaften" anklicken. "Internetprotokoll (TCP/IP)" doppelklicken. Die Einstellungen wie im Screenshot gezeigt vornehmen. Keine Einträge bei Standardgateway und DNS-Serveradressen vornehmen, diese sind hier nicht notwendig. OK übernimmt die Einstellungen.

Unter Windows Vista muss zunächst noch das Programm TFTP.EXE aus dem Verzeichnis %WINDIR%\winsxs\x86_microsoft-windows-t...-deployment-package_31bf3856ad364e35_6.0.6000.16386_none_5c9 in das Verzeichnis %WINDIR%\system32 kopiert werden. Bei älteren Windows-Versionen ist das nicht notwendig.

1. Nun wird das Senden der Firmware vorbereitet. Über Start -> Ausführen... in die Zeile "Öffnen:" **cmd** eintragen und ENTER drücken. Die Kommandozeile öffnet sich. Das TFTP-Kommando ist hier einzugeben; jetzt noch NICHT die ENTER-Taste drücken, das kommt später erst: `tftp -i 192.168.11.1 PUT (Dateiname; also generic.bin, mini.bin usw.)` Hier noch **nicht ENTER drücken**.

2. Den Stromstecker wieder in den Router einstecken. Die LEDs der Switchports fangen an, zu leuchten.
3. Im Kommandozeilenfenster **in dem Augenblick, wenn die Switch-Port LEDs nicht mehr leuchten, jetzt ENTER drücken**. War der Flashvorgang erfolgreich, kommt eine Meldung über die übertragene Datenmenge. War das Flashen nicht erfolgreich, kommt ein Timeout. Wiederholen Sie bitte die letzten beide Schritte (Strom raus, Strom rein, TFTP mit ENTER starten).
4. Nach dem erfolgreichen Flashen muss man nun 1-2 Minuten warten, da erst jetzt der eigentliche Flashvorgang im Inneren des Routers beginnt.
5. Der Router bootet nach erfolgreichem Flashen durch und steht dann unter der IP-Adresse 192.168.1.1 zur Verfügung. Sollte dies nicht der Fall sein, ziehen Sie bitte den Stromstecker ab und stecken Sie in erneut ein (der so genannte Powercycle im Englischen).
6. Setzen Sie die manuell vergebene, statische IP-Adresse zurück auf "IP-Adresse automatisch beziehen" UND setzen Sie die Auswahl bei "DNS-Server automatisch beziehen", um eine DHCP-IP-Adresse vom Router zu erhalten. Oder wechseln Sie in das 192.168.1.0/24 Netzwerk.
7. Die Weboberfläche des Routers öffnet sich über <http://192.168.1.1> mit Benutzernamen root und Passwort admin.

TFTP Anweisungen für Linux

```
tftp
tftp> binary
tftp> connect 192.168.11.1
tftp> status
Connected to 192.168.11.1.
Mode: octet Verbose: off Tracing: off
Rexmt-interval: 5 seconds, Max-timeout: 25 seconds
tftp> put /path/to/dd-wrt.v23_mini_generic.bin
```

Ein alternatives tftp Kommando für Linux (bitte entweder das Eine oder das Andere, aber nicht beide):

```
tftp 192.168.11.1 -m binary -c put dd-wrt.v23_generic.bin
```

1. Bis zur letzten Zeile eintippen (ohne Enter zu drücken!),
2. den Router einschalten und warten, bis die Switch-Port LEDs ausgehen und
3. **jetzt** <ENTER> drücken.

- Die 'status' Ausgabe ist etwas missverständlich formuliert. Der Router muss einmal vom Strom genommen und wieder angesteckt werden nach dem Flashen.

Problembhebung

- Sollten Probleme auftreten, könnte das NVRAM gelöscht werden müssen. Öffnen Sie dazu eine Telnet-Session auf den Router und führen Sie eine (oder beide?) dieser Optionen durch:
- Option 1 :

```
mtd erase nvram
```

- Option 2 : (übernommen von [OpenWrt FAQ](#))

```
cd /tmp
wget http://openwrt.inf.fh-brs.de/~nbd/nvram-clean.sh
chmod a+x /tmp/nvram-clean.sh
/tmp/nvram-clean.sh
nvram commit
```

- Suchen Sie im Forum nach Postings, die den Suchbegriff "WHR-G54S" und Ihr Problem enthalten; Englischkenntnisse sind Pflicht.
 - ◆ Wenn Ihr Router zerschossen ist, suchen Sie im Forum [forums](#) nach "WHR-G54S bricked". Es gibt ausreichend Hilfe in diesen Threads, Sie müssen es nicht auf eigenen Wegen versuchen.

Buffalo WHR-HP-G54 - ändern der Boardflags

- Telnet auf den Router
- folgendes Kommando eingeben

```
nvrn show | grep boardflags  
boardflags=0x1758
```

~ #

- erscheint 0x1758, ändern in 0x2758
- dazu dieses Kommando benutzen:

```
nvrn set boardflags=0x2758  
nvrn commit  
reboot
```

- Diese Einstellung aktiviert ein Feature auf der Empfangsseite des High Power Routers. Die Empfangsempfindlichkeit steigt extrem an.
- Dieses Feature wird nicht zusammen mit dem GUI Backup gespeichert. Es muss nach jedem Reset-to-Default neu eingestellt werden.

Wiederbeleben

Verschiedene Methoden zum [Wiederbeleben](#) eines kaputt geflashten WRT findet ihr hier:
Wiki Sektion [Wiederbeleben](#)

Das Web-Interface



Das aktuelle Standard-Interface von DD-WRT v23/v24

DD-WRT hebt sich gegenüber anderen Projekten wie zum Beispiel OpenWRT dadurch heraus, dass es eine GUI besitzt, die eine einfache Konfiguration der manigfaltigen Optionen erlaubt. So lassen sich alle Standardfunktionen von DD-WRT leicht über das Webinterface erreichen, ohne auf die Konsole zurückgreifen zu müssen (was natürlich weiterhin möglich ist). Auch wenn sich noch immer nicht alles über das web interface machen lässt, ist die komplett redesignte Variante für die DD-WRT Versionen v23/v24 ein weiterer Schritt in diese Richtung. Es unterstützt sowohl mehrere Sprachen als auch verschiedene Designs.

Man kann sich durch die verschiedenen Interfaces der verschiedenen Versionen unter folgenden Links durchklicken:

Alchemy 5a: (fast identisch mit dem Interface der DD-WRT-v22-Version):

<http://www.0711-stuttgart.de/192.168.1.23>

(Auch wenn es nicht mehr tauf frisch ist, kann man durchaus Parallelen zwischen den Benutzeroberflächen ziehen.)

DD-WRT V22finalR2: (Sorry wegen des Popups, der Webspaces ist dafür kostenlos)

<http://www.information.gmxhome.de/DDWRT/Standard/V22R2/index.html>

(dort dann unten rechts "GUI Setup" auswählen)

Version DD-WRT V23final: (Sorry wegen des Popups, der Webspaces ist dafür kostenlos)

<http://www.information.gmxhome.de/DDWRT/Standard/V23final/index.html>

Version DD-WRT V24betaVPN: (Sorry wegen des Popups, der Webspaces ist dafür kostenlos)
<http://www.information.gmxhome.de/DDWRT/Standard/V24BetaVPN/index.html>

Bei diesen Versionen kann man direkt auf die Links klicken. Die Passwortabfrage ist beim Original natürlich vorhanden.

Beschreibung der einzelnen Einstellungen

Hier findet man eine Beschreibung der einzelnen Einstellungen

Konfigurationsbeispiele

WRT54G im Client-Modus

Wiki-Beitrag

Hier außerhalb des wiki eine Musterkonfiguration (leider noch auf Basis Alchemy pre5.3) für den Betrieb des WRT54G im Client-Modus an einem Access-Point.

WRT54GL als Repeater Bridge

Mit dieser Konfiguration wird ein WRT54GL zu einer WLAN-Bridge, die ein WLAN-Signal eines beliebigen Routers weiterverteilt. Es muß nur der WRT54GL konfiguriert werden. Vom ersten Router benötigt man nur IP,SSID und gegebenenfalls den Key für die Verschlüsselung. Ich habe festgestellt, dass die Firmware V24 auf WRT54GL wohl nicht funktioniert, mit der V24 beta funktioniert es aber wunderbar. Im folgenden wird der WRT54GL als zweiter Router bezeichnet. Des weiteren gehe ich davon aus, dass der erste Router die IP 192.168.2.1 hat. Ich habe die Beschreibung aus dem englischen DDWRT-Wiki.

1. Führe einen Reset des zweiten Routers durch
2. Verbinde deinen Rechner via Kabel oder Wlan mit dem Router und logge Dich ein.
3. Öffne den **Wireless -> Basic Settings** Tab
 - ◆ Wireless Mode : Repeater Bridge
 - ◆ Wireless Network Mode : Passend zum ersten Router
 - ◆ Wireless Network Name(SSID) : Passend zum ersten Router
 - ◆ Wireless Channel : Passend zum ersten Router
 - ◆ Wireless SSID Broadcast : Enable
 - ◆ Network Configuration : Bridged
 - ◆ **Save Settings**
4. **Virtual Interfaces Section**
 - ◆ Add*[Bemerkung] kann unter Umständen zu Problemen führen. Sollte der Repeater Router den ersten Router nicht finden, diesen Schritt rückgängig machen. War z. B. so an meinem dlink DIR-600 ki

- ◆ Wireless Network Name(SSID) : Anders als der erste Router*[Bemerkung] - being a bridged setup and all DHCP is being done by the host AP - it is possible to use the same SSID as the Host AP for this Virtual SSID name. This will allow roaming hand-offs between the 2 units. [redhawk Confirmed --ChuckMcB 00:34, 31 March 2008 \(CEST\)](#)
 - ◆ Wireless SSID Broadcast : Enable
 - ◆ AP Isolation : Disable
 - ◆ Network Configuration : Bridged
 - ◆ **Save Settings**
5. Öffne den **Setup** -> **Basic Setup** Tab
- ◆ Connection Type will be: Disabled
 - ◆ Set STP for Disabled (Enabled sometimes can cause connection problems) [redhawk](#)
 - ◆ IP Address : 192.168.2.2 (Wenn die IP vom ersten Router 192.168.2.1 ist)
 - ◆ Mask : 255.255.255.0
 - ◆ Gateway: 192.168.2.1 (Wenn die IP vom ersten Router 192.168.2.1 ist)
 - ◆ Assign WAN Port to Switch : Häkchen oder nicht ->egal
 - ◆ **Save Settings**
6. Öffne den **Security** -> **Firewall** Tab
- ◆ Alle Häkchen wegmachen
 - ◆ Disable SPI firewall
 - ◆ **Save Settings**
7. Öffne den **Administration** Tab
- ◆ **APPLY Settings**
8. Dann noch bitte unter "Setup"-> "Netzwerke" bei Punkt "DHCPD" einen multiblen DHCP Server hinzufügen!***APPLY Settings**
9. Am Client noch DHCP einstellen, so hat es bei mir funktioniert! Viel Erfolg! Mfg!

Nun solltest Du in der Lage sein, kabellose und kabelgebunden Clients, an den neu konfigurierten zweiten Router anzuschließen. Diese erhalten ihre IP-Adressen vom ersten Router und sollten in der Lage sein eine Internetverbindung über den ersten Router aufzubauen.[turol 23:59, 11 June 2008 \(CEST\)](#)

Verschlüsselung des Virtuellen Interfaces

Solltet ihr den Wunsch verspüren das Virtuelle Interface verschlüsseln zu wollen, müsst ihr darauf achten das das Virtuelle und das Physische Interface absolut identische Verschlüsselungs-Einstellungen haben, auch die Passwörter müssen identisch sein!

WRT an einer FritzBox (o.a.) für ein Nachbarschaftsnetz

[Hier ein extra Wiki-Beitrag](#)

DD-WRT Router als WDS einrichten um ein Netz mit mehreren Routern aufzubauen, an denen sich Clients anmelden können

Mit Hilfe von WDS (Wireless Distribution Service) kann ein Netzwerk von mehreren (WLAN) Routern eingerichtet werden, die alle unter der gleichen SSID erscheinen und somit ein "großes" Netz darstellen. Es handelt sich hier nicht um die "Bridging"-Funktion, es ist hier im Gegensatz dazu möglich, daß sich **WLAN Clients am gesamten Netz anmelden können** und die Router nicht nur (wie eben beim Bridging) eine direkte Verbindung zum Überwinden größerer Distanzen herstellen sollen. Das WLAN Netzwerk kann so nahezu beliebig erweitert werden, wobei der Anwender nicht mehrkt, in welchem der angeschlossenen Router er sich befindet (ausser er möchte es wissen).

Ich habe Informationen gesammelt, um dieses Vorhaben mit Linksys WRT54x Routern und DD-WRT zu realisieren, ich weiß nicht, ob dies auch mit anderen Routern und DD-WRT funktioniert.

Vorbereitungen

Im Vorfeld ist ein

DD-WRT-Router als **Internet-Router**, der eben mit dem Internet verbunden ist und eine beliebige Zahl von

DD-WRT-Routern als **WDS-Router** zu bestimmen, die das WLAN-Netz des **Internet Routers** erweitern. Diese Unterscheidung ist wichtig, da es verschiedene Einstellungen für den **einen Internet-Router** und die **vielfahl WDS-Routern** gibt.

Falls notwendig, könnt Ihr die Konfiguration Eures DD-WRT Routers absichern, um diesen Zustand später wieder zu erreichen (**Administration -> Backup**). Ich werde im übrigen die englischen Navigationspunkte verwenden, daß Ihr ggf. diese Anleitung mit anderen aus dem Internet verwenden könnt.

Ich sehe es weiterhin als ideal an, wenn Ihr Eure Router (ja, alle!) auf die Werkseinstellungen zurücksetzt (**Administration -> Factory Default - Klick auf "yes" - und "Save Settings"**). **Jeder Router ist jetzt unter der TCP/IP (im Browser eingeben) 192.168.1.1 und den Zugangsdaten root/admin erreichbar.**

WICHTIG: Nur der Hauptrouter darf mittels Netzkabel mit dem internen Netz verbunden sein. Die anderen Router nehmen untereinander per WLAN zueinander Verbindung auf.

Einstellungen in DD-WRT

I. Folgende Daten sind im Vorfeld zu notieren

1. (**Wireless -> WDS**) Die MAC Adressen Eurer Router auf, die Ihr am Beginn dieser Seite findet. **Verwendet unbedingt die MAC Adressen, wie Ihr sie auf der "WDS" Seite ablesen könnt, denn diese Adresse kann von der, die Ihr auf der Rückseite oder auf der Packung Eures Routers findet abweichen!**
2. (**Status -> Router -- Internet**) am Router, der (tatsächlich aber testweise) mit dem Internet verbunden ist notiert Ihr die **ISP-DNS** 1-3.

DD-WRT Router als WDS einrichten um ein Netz mit mehreren Routern aufzubauen, an denen sich Clients

II. Für alle Router (Den einen Internet- und die Vielzahl von WDS- Routern) gleich

1. (**Setup -> Basic Setup**) Hier wird jedem Router eine andere TCP/IP Adresse zugewiesen. Also z.B. 192.168.1.1, 192.168.1.2, 192.168.1.3 usw.. Weiterhin wird dem Router unter **Router name** ein eindeutiger Name zugewiesen, z.B. WDS-1, WDS-2, WDS-3..usw.
2. (**Setup -> Basic Setup -- Network Address Server Settings (DHCP) -- Start IP Address**) Es ist notwendig festzulegen, ab welcher TCP/IP Adresse jeder Router über DHCP die Adressen an die Clients vergibt. So könnt Ihr feststellen, über welchen Router im gesamten Netz ein Client angemeldet ist. Ich habe bspw. dem Router mit der TCP/IP 192.168.1.1 eingetragen, daß er mit 192.168.1.100 beginnt. Der mit 192.168.1.2 mit 192.168.1.200, bitte beachten bis max *.254... verstanden? Dieser Schritt ist zudem wichtig, daß in Eurem Netz keine TCP/IP Adressen doppelt vergeben werden.
3. (**Wireless -> Wireless Security -> Security Mode -> Disabled**) Deaktiviert das WLAN Kennwort Eurer Router. Wenn alles funktioniert könnt (und solltet) Ihr es später aktivieren.
4. (**Wireless -> Basic Settings**) Hier gibt es einiges einzustellen. Bei allen Routern muss **AP mode** angewählt werden und es muss der gleiche **channel** für alle Router gewählt werden. Dann wird für alle Router ein SSID (WLAN Name) gewählt.
5. (**Wireless -> WDS**) Von dieser Seite habt Ihr bereits die MAC Adressen abgeschrieben. Jetzt tragt Ihr in die vorgegebene Liste darunter die MAC Adressen **aller anderen** Router (die Ihr bereits notiert habt) und wählt unter "**Type**" LAN. Als Namen habe ich den Namen des Routers, wie unter (**Setup -> Basic Setup -- Router name**) - siehe Schritt 1 - festgelegt, verwendet. Lasst die Pfoten von **Lazy WDS und WDS subnet** bzw. lasst die auf **Disabled**.

III. Für den Internet-Router

Hier gilt es nur zu testen, ob der Router auch die Internetverbindung herstellt, dazu muss regelmäßig unter (**Setup -> Basic Setup -> Internet Setup -> Internet Connection Type ->**) DHCP gewählt sein, aber dass hängt von Eurem Internet bzw. DSL- oder Kabelmodem ab. Probiert es einfach mit DHCP. Probiert an dieser Stelle, ob Ihr euch an diesem Router anmelden könnt und ob das Internet verfügbar ist. --> Auf Eurem Laptop (wenn Ihr leider keinen Apple haben solltet) muss DHCP aktiviert sein.

IV. Für alle anderen (WDS) Router

1. (**Setup -> Basic Setup -> Internet Setup -> Internet Connection Type -> Disable**)
2. (**Setup -> Basic Setup -> Network Setup -> Router IP -> Gateway**) Hier die oben unter II. 1. festgelegte TCP/IP Adresse des **Internet-Routers** eingeben.
3. (**Setup -> Basic Setup -- Network Address Server Settings (DHCP) -- Static DNS 1,2,3**) Hier die unter I. 2. notierte ISP DNS eintragen.
4. (**Security -> Firewall -> Firewall Protection -> Disable**)

V. Testen

Schaltet die Router alle an und gönnt ihnen einige Zeit um die WDS Verbindungen aufzubauen. Die neuen Linksys WRT54x quittieren diese Vorgang mit der orangenen LED an der Vorderseite. Zur Kontrolle könnt Ihr auch die einzelnen Router (über die unter II. 1. festgelegten) TCP/IP Adressen pingen. Weiterhin sehr Ihr auf der Statusseite der Router (die Ihr natürlich auch über die in II. 1. festgelegte TCP/IP erreicht), ob sich die WDS Router gegenseitig "finden".

Jetzt wird es zeit unter (**Administration -> Router Management**) ein Passwort für das DD-WRT Administrationsprogramm zu vergeben (anstatt: root/admin).

Linux inside

Bekanntlich ist die DD-WRT-Firmware ein Mini-Linux, d. h. neben allem was man mit der Web-Oberfläche mittels Browser einstellen kann gibts auch noch eine Linux-Shell mit noch mehr Möglichkeiten und Befehlen. Zugriff darauf bekommt man (auch unter Windows) nachdem man unter administration -> management enabled, mittels telnet oder SSH, also z. B. in die Eingabeaufforderung

telnet 192.168.1.1 (richtige IP-Adresse des Routers einsetzen)

eingeben. Login/Passwort sind standardmäßig root/admin oder je nachdem was man sonst vorher eingestellt hat. Für SSH muss man unter Windows ein SSH-Programm verwenden, z. B. Putty, Terraterm usw. nach dem gleichen Muster.

Momentan läuft die Firmware mit Kernel 2.4.20. Das kann man sich in einer Telnet-Sitzung ausgeben lassen mit dem Befehl 'cat /proc/version'.

Linux version 2.4.20 (root@colinux) (gcc version 3.3.5) #766 Thu Jun 23 16:30:31 UTC 2005

Unter der Shell befindet sich auch noch eine komplette Verzeichnisstruktur mit einzelnen Ordner und Dateien.

Wichtige Linuxbefehle der Shell

Grundsätzlich gibt die Eingabe von 'help' die eingebauten, also von der Shell selbst bereitgestellten Befehle aus, zusätzlich kann man sich noch mittels zweimaligen Drückens der Tab-Taste eine Vielzahl weiterer Befehle ausgeben lassen.

Traffic control (traffic shaping) mit tc

Manchmal möchte man den Verkehr auf bestimmten Ports (z. B. FTP-Port 21 oder P2P-Ports) drosseln, damit diese zwar laufen, aber noch ausreichend Bandbreite für andere Zwecke übrig bleibt. Da der WRT54G unter Linux läuft, kann man auch Standard-Linux-Traffic-Shaping-Tools wie tc (traffic control) einsetzen. Leider ist die Bedienung mit den vielen Parametern sehr kryptisch. Das Samba-Share-Laufwerk der DD-WRT-Firmware kann dabei das Laden des Shell-Skripts vereinfachen. Im Berliner Quake-Forum gab es folgendes Beispiel, welches Port 1234 auf 8 kBit/s begrenzt und einen Eindruck der tc-Parameter gibt:

```
tc qdisc add dev eth0 root handle 1: cbq avpkt 1000 bandwidth 10mbit
tc class add dev eth0 parent 1: classid 1:1 cbq rate 8kbit allot 1500 prio 5 bounded isolated
tc filter add dev eth0 parent 1: protocol ip prio 16 u32 match ip sport 1234 0xffff flowid 1:1
tc qdisc add dev eth0 parent 1:1 sfq perturb 10
```

Weiterführende Links:

- ◆ [Linux Advanced Routing & Traffic Control \(lartc\)](#) (engl.)
- ◆ [TC: a tool for Routing & Traffic Control](#)
- ◆ [Traffic Shaping with Linux](#)
- ◆ [Das WonderShaper-Skript](#)

selber compilieren

[DD-WRT compilieren](#)

Skript-Schnipsel

Eine kleine Sammlung von nützlichen [Shell-Skripten](#)

Antennen ausrichten

Client-Modus

Um die Antennen Richtung Access-Point genau auszurichten, kann man sich die Signalstärke seines AP kontinuierlich ausgeben lassen. RSSI sollte möglichst hoch sein (-63 dB ist besser als -86 dB), Noise möglichst gering (-91 dB ist besser als -81 dB), die Differenz zwischen beiden (Signal-Noise-Ratio (SNR)) sollte mindestens 10 dB betragen.

Man startet eine Telnet-Sitzung zum Router und fügt das folgende Skript mit Copy & Paste ein:

```
while [ 1 ];
do CH=`wl channel|grep target|cut -c16-`;
MAC=`wl assoclist|cut -d' ' -f2`;
RA=`wl rate|cut -d' ' -f3`;
RSSI=`wl rssi|cut -d' ' -f3`;
NOISE=`wl noise|cut -d' ' -f3`;
let SNR="$RSSI- $NOISE";
echo "MAC=$MAC Kanal $CH: Rate=$RA RSSI=$RSSI Noise=$NOISE SNR=$SNR";
sleep 1;
done
```

Das Skript führt in einer Endlosschleife nacheinander die Befehle *wl channel*, *wl assoclist*, *wl rate*, *wl rssi*, *wl noise* aus, formatiert das Ergebnis mittel *grep* und *cut* zu der Ausgabezeile und wartet dann 1 s bis zur nächsten Wiederholung. Da die Formatierungsbefehle auf die Firmware abgestimmt sind, funktioniert das Skript nicht mit x-beliebigen Firmwares (hier getestet mit DD-WRT#22pre5). Das Ergebnis sieht dann beispielsweise so aus (knapp eine Zeile pro Sekunde):

DD-WRT_Doku_(DE)

```
MAC=00:0B:6B:32:A7:83 Kanal 10: Rate=11 RSSI=-63 Noise=-91 SNR=28
MAC=00:0B:6B:32:A7:83 Kanal 10: Rate=11 RSSI=-64 Noise=-91 SNR=27
MAC=00:0B:6B:32:A7:83 Kanal 10: Rate=11 RSSI=-64 Noise=-90 SNR=26
```

Anstelle der MAC kann man sich auch die SSID ausgeben lassen, mit folgendem Skript:

```
while [ 1 ];
do CH=`wl channel|grep target|cut -c16-`;
SID=`wl ssid|cut -c15-`;
RA=`wl rate|cut -d' ' -f3`;
RSSI=`wl rssi|cut -d' ' -f3`;
NOISE=`wl noise|cut -d' ' -f3`;
let SNR="$RSSI- $NOISE";
echo "SSID=$SID Ch $CH: Rate=$RA RSSI=$RSSI Noise=$NOISE SNR=$SNR";
sleep 1;
done
```

```
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-63 Noise=-90 SNR=27
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-64 Noise=-91 SNR=27
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-63 Noise=-90 SNR=27
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-62 Noise=-91 SNR=29
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-63 Noise=-91 SNR=28
```

AP-Modus

Um das Skript im AP-Modus zu nutzen, um die Antenne Richtung Client auszurichten, muß man beim "wl rssi"-Befehl die MAC-Adresse eines Clients angeben und den Befehl "wl assoclist" und die MAC-Ausgabe entfernen.

```
while [ 1 ];
do MAC=00:0B:6B:32:A7:83;
CH=`wl channel|grep target|cut -c16-`;
RA=`wl rate|cut -d' ' -f3`;
RSSI=`wl rssi $MAC|cut -d' ' -f3`;
NOISE=`wl noise|cut -d' ' -f3`;
let SNR="$RSSI- $NOISE";
echo "MAC=$MAC Kanal $CH: Rate=$RA RSSI=$RSSI Noise=$NOISE SNR=$SNR";
sleep 1;
done
```

Wenn die MAC-Adresse eines bekannten Clients bei 00:0B:6B:32:A7:83 eingesetzt wird, ergibt sich folgendes:

```
MAC=00:0B:6B:32:A7:83 Kanal 10: Rate=11 RSSI=-63 Noise=-91 SNR=28
MAC=00:0B:6B:32:A7:83 Kanal 10: Rate=11 RSSI=-64 Noise=-91 SNR=27
MAC=00:0B:6B:32:A7:83 Kanal 10: Rate=11 RSSI=-64 Noise=-90 SNR=26
```

das 2. Beispiel umgeschrieben, auch wieder mit einer bekannten MAC-Adresse eines Clients (hier: 00:0B:6B:32:A7:83)

```
while [ 1 ];
do MAC=00:0B:6B:32:A7:83;
CH=`wl channel|grep target|cut -c16-`;
SID=`wl ssid|cut -c15-`;
RA=`wl rate|cut -d' ' -f3`;
RSSI=`wl rssi $MAC|cut -d' ' -f3`;
```

```
NOISE=`wl noise|cut -d' ' -f3`;
let SNR="$RSSI- $NOISE";
echo "SSID=$SSID Ch $CH: Rate=$RA RSSI=$RSSI Noise=$NOISE SNR=$SNR";
sleep 1;
done
```

```
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-63 Noise=-90 SNR=27
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-64 Noise=-91 SNR=27
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-63 Noise=-90 SNR=27
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-62 Noise=-91 SNR=29
SSID="SiROTTER" Ch 10: Rate=11 RSSI=-63 Noise=-91 SNR=28
```

Getestet im WDS-Betrieb mit:

2 x WRT54GSv4

2 x WRT45GL

Software installieren mit ipkg

IPKG ist eine Debian-ähnliche Paketarchitektur, die die Installation von Software sehr vereinfacht.

Port-Knocking

Dieser Abschnitt stellt ein kleines How-To zur Installation und Konfiguration des OpenWRT-Paketes knockd dar. Der Autor erhebt keinen Anspruch auf Vollständigkeit oder Perfektion, aber da es sich sowieso um ein Wiki handelt lasst eurer Kritik freien Lauf und ändert was ihr für nötig haltet ;-) by Age_M

Was ist Port-Knocking?

Port-Knocking hat, wie der Name bereits impliziert, etwas mit Klopfzeichen zu tun. Die meisten kennen sicherlich den berühmten Ausruf "Sesam, öffne dich!" aus 'Ali Baba und die 40 Räuber', mit dem sich Ali Baba den Zutritt zur Höhle der 40 Räuber verschafft hat. Ähnlich funktioniert auch das Port-Knocking, allerdings nicht durch Sprache, sondern durch eine vorher definierte Sequenz aus TCP- und UDP-Paketen. Ein Daemon, der 'knockd', überwacht den eingehenden Netzwerkverkehr einer Schnittstelle und überprüft diesen auf die in der Konfigurationsdatei festgelegten Sequenzen. Wird eine Sequenz erkannt, zum Beispiel ein TCP-Paket auf Port 6666, dann ein UDP-Paket auf Port 4444 und schließlich wieder ein TCP-Paket auf Port 6666, so wird der in der Konfigurationsdatei festgelegte Befehl ausgeführt. Die Befehle sind dabei vom Benutzer frei wählbar.

Wozu braucht man Port-Knocking?

Port-Knocking kann in vielerlei Hinsicht eingesetzt werden. Eine Anwendungsmöglichkeit ist zum Beispiel die Abwehr von Portscans. Da Portscanner häufig einen Port nach dem anderen durchprobieren, können diese erkannt und wirksam durch das dynamische Anpassen der Firewallkonfiguration oder durch eine Trennung der Internetverbindung vereitelt werden. Eine weitere Anwendungsmöglichkeit ist die Fernadministration. Der Systemadministrator ist mit Port-Knocking in der Lage einzelne Befehle oder Skripte auszuführen ohne das

ein direkter Zugriff auf den Host benötigt wird. Am häufigsten jedoch wird Port-Knocking dazu verwendet, Dienste wie zum Beispiel SSH, HTTP oder FTP zu verstecken und so das Angriffsrisiko zu minimieren. Dafür wird ebenso wie bei der Abwehr von Portscans das dynamische Anpassen der Firewallkonfiguration verwendet.

Voraussetzungen

```
-WRT54G/GS oder DD-WRT-kompatibles Gerät (Autor verwendet: WRT54GS_v1.1)
-DD-WRT_Firmware (Autor verwendet: DD-WRT_v23_alpha_VoIP_2005-08-22)
-JFFS muss aktiviert sein und noch über mindestens 700 kB Speicher verfügen
```

Bei dd-wrt-v23-sp2 vom 30.08.06 scheint es Probleme bei der Aktivierung von jffs2 per Web-GUI zu geben. Die manuelle Variante über eine ssh-Konsole funktioniert allerdings. [Jffs](#) hilft weiter.

Installation der benötigten Pakete

Zunächst muss eine Remote-Shell via SSH oder Telnet auf dem Router geöffnet und die Paketdatenbank von 'ipkg' aktualisiert werden. Dies geht mit dem Befehl:

```
~# ipkg update
```

Das benötigte Paket 'knockd' findet man über den [OpenWRT-Package-Tracker](#). Alternativer Download über: <http://voidmain.is-a-geek.net/files/ipkg/>. Installiert wird es mit dem Befehl:

```
~# ipkg install http://openwrt.alphacore.net/knockd\_0.4\_mipsel.ipk
```

Nach der installation von 'knockd' fehlt nun noch das Paket 'libpcap' welches mit dem Befehl

```
~# ipkg install libpcap
```

installiert wird. Alle installierten Dateien liegen nun unterhalb von /jffs/ und die Konfigurationsdatei 'knockd.conf' befindet sich in /jffs/etc/. Als nächstes sollte die Funktion von 'knockd' mit dem Befehl

```
~# knockd --help
```

überprüft werden. Sollte es hierbei zu der Fehlermeldung "knockd: can't load library 'libpcap.so.0.8'" kommen, müssen lediglich die Pfade für die Librarys erneut exportiert werden. Dies geht mit dem Befehl:

```
~# export LD_LIBRARY_PATH=/lib:/usr/lib:/jffs/usr/lib
```

Sollte eine neuere Version von libpcap.so installiert sein, so hilft ein symbolischer Link darauf:

```
ln -s libpcap.so.0.9 libpcap.so.0.8
```

Dies sollte natürlich im entsprechenden Verzeichnis /jffs/usr/lib ausgeführt werden. Nun sollte eine erneute Funktionsüberprüfung von knockd zum Erfolg führen.

Startskripte erstellen

Zunächst sei gesagt, das die ersten zwei Schritte nicht zwingend notwendig, jedoch der Übersichtlichkeit dienlich sind.

Erstellen eines Ordners für Skripte

In diesem Fall /jffs/scripts.

```
~# mkdir /jffs/scripts
```

Erstellen eines allgemeinen Startup-Skripts

Dieses kann wiederum weitere Skripte starten.

```
~# echo "#!/bin/sh" > /jffs/scripts/startup.sh
~# echo "# Insert startup-scripts here" >> /jffs/scripts/startup.sh
```

Erstellen eines Skriptes für den Start von knockd

Als nächstes wird das Skript für den Start von knockd erstellt.

```
~# echo "#!/bin/sh" > /jffs/scripts/start_knockd.sh
~# echo "export LD_LIBRARY_PATH=/lib:/usr/lib:/jffs/usr/lib" >> /jffs/scripts/start_knockd.sh
~# echo "knockd -d -i ppp0 -c /jffs/etc/knockd.conf" >> /jffs/scripts/start_knockd.sh
```

Die Verwendung des Parameters '-i' ist von der zu verwendenden Schnittstelle abhängig. Bei einer DSL-Verbindung mit PPPoE lautet diese Schnittstelle 'ppp0'. Wird der Parameter weggelassen, wird die Schnittstelle 'eth0' verwendet.

Da die Skripte bisher nur reguläre Dateien sind, müssen diese noch ausführbar gemacht werden.

```
~# chmod 755 /jffs/scripts/*
```

Automatischen Start einrichten

Damit 'knockd' automatisch beim Bootvorgang des Routers ausgeführt wird, muss die variable rc_startup des NVRAM noch überarbeitet werden. Wenn die Schritte 5.1 und 5.2 übersprungen wurden lautet der Befehl:

```
~# nvram set rc_startup="/jffs/scripts/start_knockd.sh"
~# nvram commit
```

Bei der Verwendung des allgemeinen Startskriptes müssen folgende Befehle verwendet werden:

```
~# nvram set rc_startup="/jffs/scripts/startup.sh"
~# nvram commit
~# echo "/jffs/scripts/start_knockd &" >> /jffs/scripts/startup.sh
```

Sollte das Script beim Start nicht ausgeführt werden, muss vorher eine kleine Pause eingelegt werden (evtl. ist

das jffs-Dateisystem noch nicht gemountet):

```
~# nvram set rc_startup="sleep 30; /jffs/scripts/startup.sh"
~# nvram commit
```

knockd.conf anpassen

Das Anpassen der knockd.conf ist schnell erledigt. Hierfür verwendet man am besten den Editor 'vi', da dieser Bestandteil der DD-WRT-Firmware ist. Also:

```
~# vi /jffs/etc/knockd.conf
```

Unter [options] braucht man lediglich die vorhandenen Optionen löschen und folgendes eintragen:

```
[options]
UseSyslog
```

Damit ist die allgemeine Konfiguration des Dienstes abgeschlossen, jedoch fehlen nun noch die Anwendungen.

Anwendungsbeispiele

Da es sich hier um ein Wiki handelt, hoffe ich, dass jeder, der interessante Anwendungsbeispiele für das Port-Knocking findet, diese hier mit kurzer Beschreibung aufführt. Informationen zur Konfiguration der Anwendungsmöglichkeiten gibts [HIER](#). Viel Spass beim "Klopfen" ;)

Beispiel 1: Hiermit könnt ihr den SSH-Port zum Router öffnen. Der Verbindungsaufbau zum Port 22 wird nur für die IP-Adresse des anklopfenden Host (dafür ist die Variable %IP% zuständig) akzeptiert und nach 30 Sekunden wieder verweigert. Da SSH allerdings das verbindungsorientierte Protokoll TCP verwendet, wird die bereits aufgebaute Verbindung nicht geschlossen, es können lediglich keine weiteren Verbindungen aufgebaut werden.

```
[opencloseSSH]
sequence = 33333:udp,22222:tcp,33333:udp
seq_timeout = 5
tcpflags = SYN
start_command = /usr/sbin/iptables -A INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
cmd_timeout = 30
stop_command = /usr/sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
```

Beispiel 2: Im Prinzip das gleiche wie Beispiel 1, aber anstelle des SSH-Ports wird der HTTPS-Port geöffnet.

```
[opencloseWEB]
sequence = 44444:udp,33333:tcp,44444:udp
seq_timeout = 5
tcpflags = SYN
start_command = /usr/sbin/iptables -A INPUT -s %IP% -p tcp --dport 443 -j ACCEPT
cmd_timeout = 30
stop_command = /usr/sbin/iptables -D INPUT -s %IP% -p tcp --dport 443 -j ACCEPT
```

Beispiel 3: Im Prinzip auch das gleiche wie Beispiel 1, diesmal wird allerdings der SSH-Port auf einen Rechner im Lan geleitet. (die beiden iptables-Befehle in start_command und stop_command gehören bei der Konfiguration natürlich in eine Zeile)

```
[opencloseSSH]
sequence = 33333:udp,22222:tcp,33333:udp
seq_timeout = 5
tcpflags = SYN
start_command = /usr/sbin/iptables -t nat -I PREROUTING 4 -s %IP% -p tcp --dport 22 -j DNAT --to
                /usr/sbin/iptables -I FORWARD 10 -p tcp -m tcp -s %IP% --dport 22 -j ACCEPT
cmd_timeout = 30
stop_command = /usr/sbin/iptables -t nat -D PREROUTING -s %IP% -p tcp --dport 22 -j DNAT --to 19
                /usr/sbin/iptables -D FORWARD -p tcp -m tcp -s %IP% --dport 22 -j ACCEPT
```

Hinweis zu beiden Beispielen:

```
/usr/sbin/iptables -A ...
```

setzt eine Filterregel ans Ende der iptables. Wenn zuvor eine DROP-Regel für den eingefügten Dienst steht, so geht diese nachgelagerte ACCEPT-Regel verloren! Um dies zu umgehen sollten die Befehle mit

```
/usr/sbin/iptables -I ...
```

an den Anfang der iptables eingefügt werden.

Die Client-Tools

Die Verwendung von 'knockd' alleine reicht natürlich nicht aus. Die Client-Seite muss ja schließlich irgendwie anklopfen können. [HIER](#) bekommt man die nötigen Tools um erfolgreich Klopfzeichen senden zu können. Wer wie der Autor 'Gentoo-Linux' auf der Clientseite verwendet, kann einfach

```
emerge knock
```

verwenden, um das benötigte Paket zu installieren. Um anschliessend eine Verbindung aufbauen zu können braucht man einfach nur folgende Befehle ausführen

```
knock <router-wan-ip> -u 44444
knock <router-wan-ip> 33333
knock <router-wan-ip> -u 44444
```

um die Sequenz von Beispiel 2 zu generieren. Der Parameter '-u' generiert dabei UDP-Pakete, wenn dieser weggelassen wird, werden TCP-Pakete generiert. Ein erfolgreiches Anklopfen stellt sich im Router-Log folgendermaßen dar:

```
knockd: 70.240.222.241 (p642CF4.dip.t-dialin.net): opencloseWEB: Stage 1
knockd: 70.240.222.241 (p642CF4.dip.t-dialin.net): opencloseWEB: Stage 2
knockd: 70.240.252.241 (p642CF4.dip.t-dialin.net): opencloseWEB: Stage 3
knockd: 70.240.222.241 (p642CF4.dip.t-dialin.net): opencloseWEB: OPEN SESAME
```

Zumindest für die Windows-Version des knock-clients funktioniert auch die folgende Semantik:

```
knock <router-wan-ip> 44444:udp 33333:tcp 44444:udp
```

Quellen

<http://wrt-wiki.bsr-clan.de/index.php?title=Knockd>

<http://www.zeroflux.org/cgi-bin/cvstrac.cgi/knock><http://wiki.dd-wrt.com/wiki>

<http://www.portknocking.org/>

Bootvorgang

Viele interessante Details zum Bootvorgang eines WRT54G v2.2

Hardware

Autopsy: Linksys WRT54G/GS Hardware Versions Under the Knife (auf englisch) gibt ein detaillierten Überblick über die Hardware und das Innenleben der einzelnen Versionen der WRT54G und GS Router einschließlich Bilder.

Modifikationen

Dual Serial Port

Einbau eines Dual-Serial-Ports in Linksys WRT54G/WRT54GS. Es gibt verschiedene Realisierungen des Vorhabens:

- ◆ Intern mit Konsole
- ◆ Extern mit LEDs (auf deutsch)

SD-Card Reader

Den Router mit einem SD-Card Reader ausstatten:

- ◆ SD-Card Reader WRT54G v2.2 und 3.1 (deutsch, Fotos)
- ◆ kiel.kool.dk (auf Englisch) (down *hat wer ein Backup?*)
- ◆ Tutorial: Hinzufügen eines SD-Kartenlesers an den Linksys WRT54G (deutsch)
- ◆ Tutorial: SD-Card Reader im WRT45GS v4/GL (deutsch)

Ab der V23 SP1 von DD-WRT ist kein zusätzliches manuelles Aktivieren des MMC-Treibers notwendig. Einfach unter Administration - Management - MMC/SD Card Support - Enable MMC Device

<http://cascade.dyndns.org/~datagarbage/dd-wrt-enable-mmc.png>

JTAG-Adapter

Anleitung für einen JTAG-Adapter zum Wiederbeleben eines kaputt geflashten WRT.

WRT350N Serial Port

The pinout of the serial is the same as for most linksys stuff... The wrt54g-series have it that way, or if the have two of them they are just doubled up...

You will still need a level shifter from TTL Level to PC-Port-Serial, but that is common for such devices. Look around for information about adding a serial port to an WRT54G, it is exactly the same stuff you need...

If you get it then open serial connection and press Ctrl+C during bootup, you will end up in CFE. there you will have to

```
nvrans set boot_wait=on  
nvrans commit
```

and see if it gets pingable again during boot (only for 3seconds, so like always, connect it to a switch, ensure your pc has got 192.168.1.X as ip and do a constant ping [normal in most *nix systems, -t for windoze]) if it is then try uploading firmware the normal tftp-way. If it ends up in some errors in the serial console about the

flash not being verified than try

```
flash -noheader : flash1.trx
```

and tftp imidiatly after typing it...

Worked for me.

WRT350N-Antennen Mod

Anleitung für einen Antennen-Mod an einem Linksys WRT350N. Achtung dies ist kein Anfänger-Mod. Beim öffnen des Gerätes verliert ihre eure Garantie. Ich übernehme keine Haftung oder sonstige Gewährleistungen für defekte Geräte. Ihr macht diesen Mod auf eigene Verantwortung.

Wie öffne ich einen Linksys WRT350N?

Anleitung: [How to open a WRT350N](#)

Allgemeine Informationen:

[Forum-Thread](#)

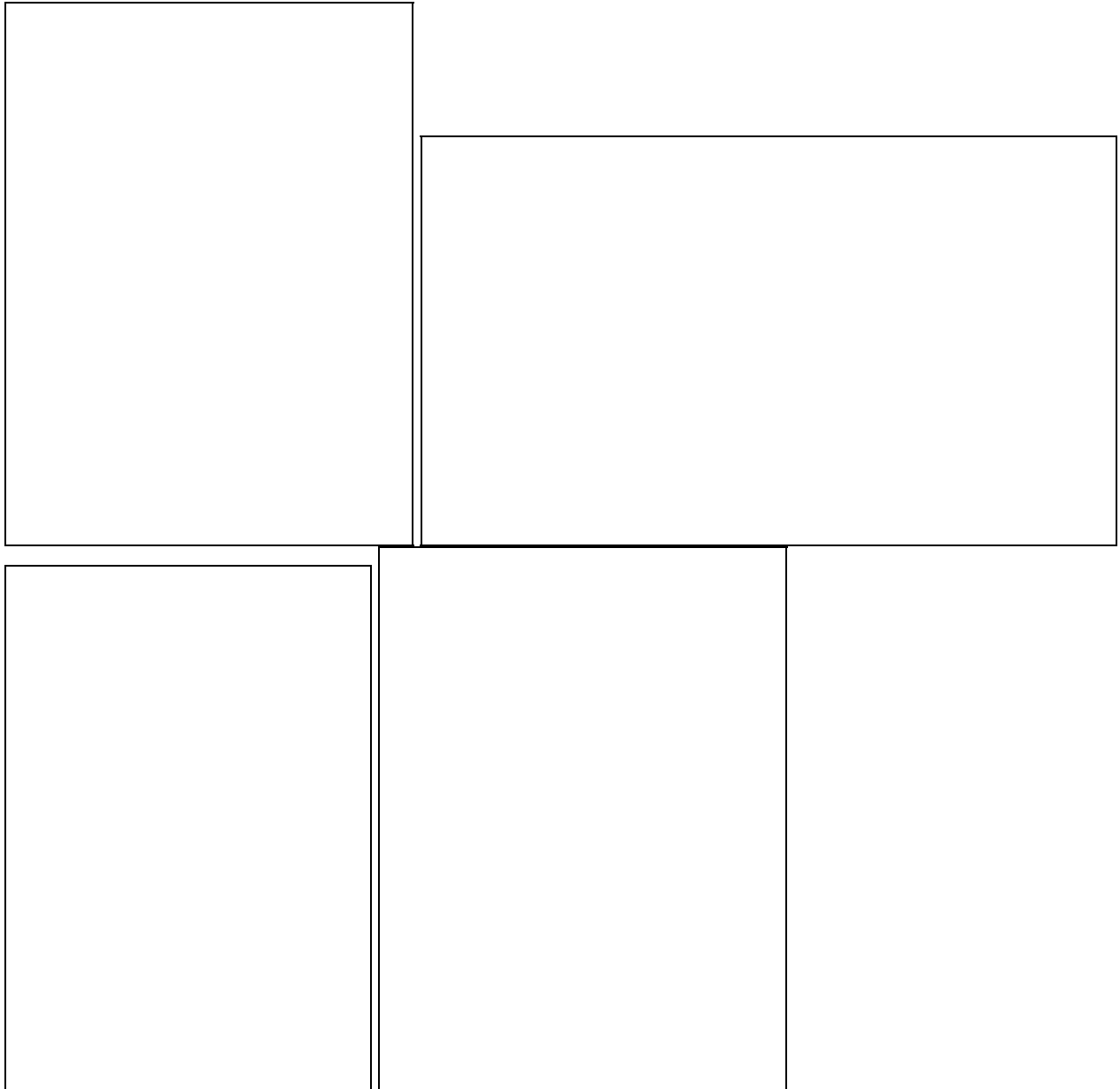
[Fotos](#)

Was bringt der Mod:

Größere Antennen bringen **längere Reichweite** und ein **besseres Signal**.

Achtung bitte beachten Sie die maximale Sendeleistung von 100 mW, dies ist in Deutschland die erlaubte Obergrenze für nicht anmeldepflichtige Funk-Hardware im 2,4-GHz-Frequenzband.

Wie es am Ende aussehen sollte:

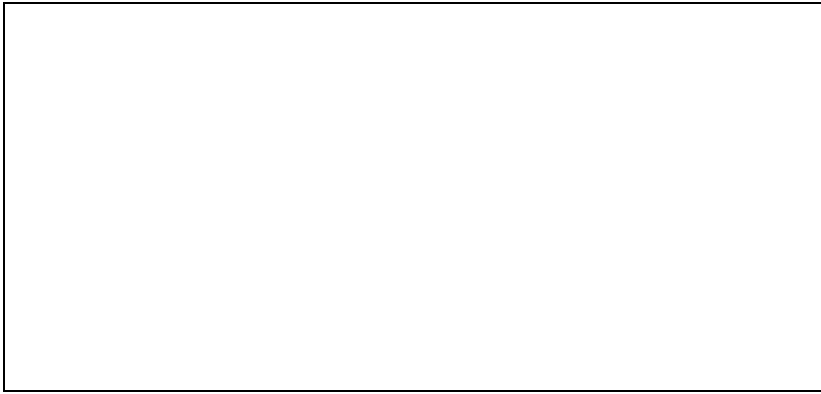


Was wird benötigt?

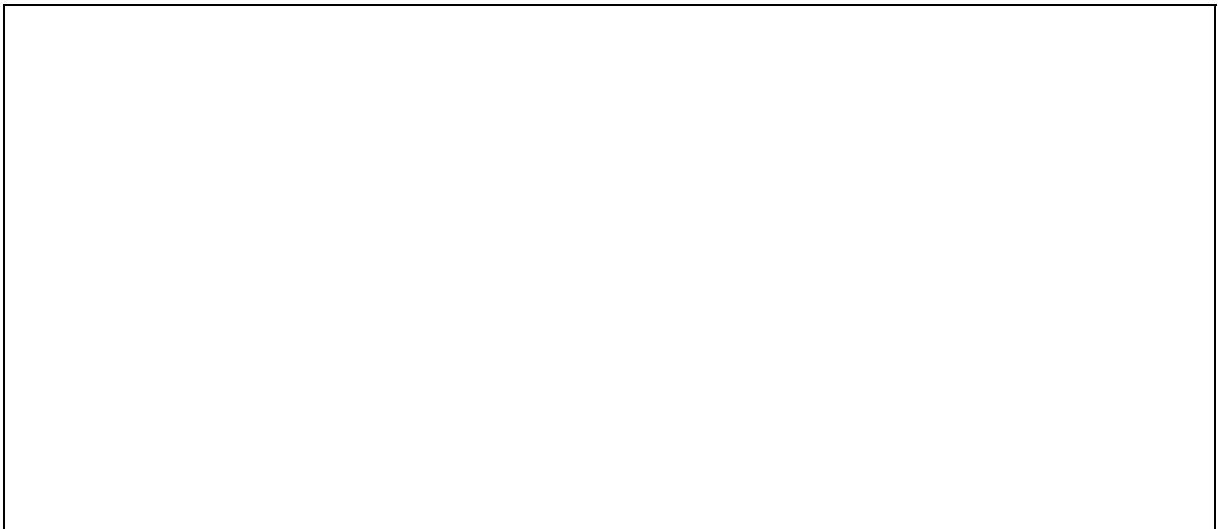
Schraubenzieher, LötKolben, 2x passende Pigtails.

Ich hab ein U.FL auf --> RP-SMA (männlich) Pigtail benutzt. Das U.FL Ende des Pigtails wurde abgeschnitten/abisoliert der U.FL Stecker ist wertlos uns interessiert nur der "Draht" und das andere RP-SMA Ende.

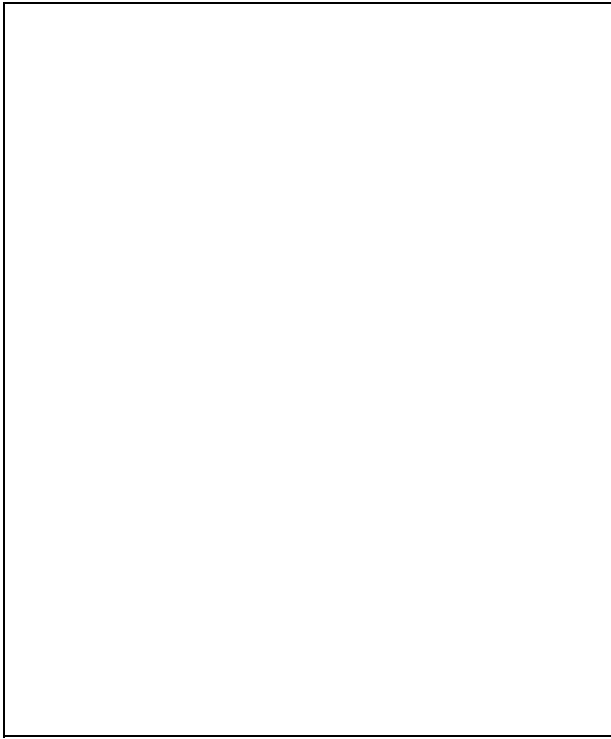
Die Idee ein Pigtail zu finden welches ihr direkt auf die Platine anstecken könnt, könnt ihr euch gleich wieder aus dem Kopf schlagen. Es handelt sich zwar um einen Hirose MS-156NB Anschluss (danke an [adisor19](#)), aber dieser ist nicht geeignet um dort ein Pigtail anzuschließen.



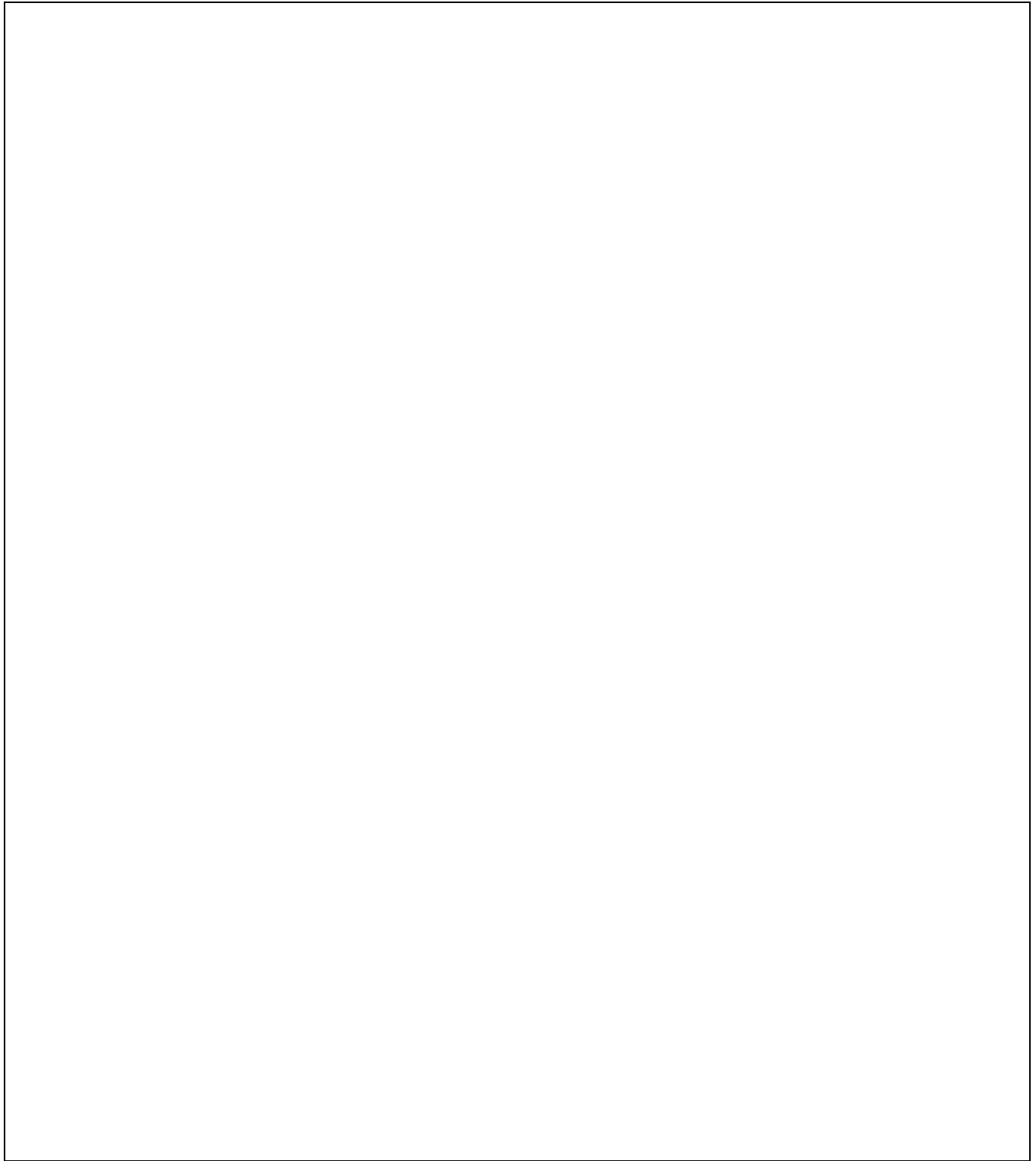
Deswegen brauchen wir einen LötKolben. Hier seht ihr die Anschlüsse, an den rot markierten Punkten setzen wir an :)



Beim rauslöten des alten Pigtails bitte aufpassen, eine Halterung zwischen welcher das Pigtail am Lötspot liegt hält das Pigtail fest/es umklammert. Einzige Möglichkeit ist das Pigtail nach hinten raus zu ziehen, nachdem wir den gesamten Punkt erwärmt haben.



Jetzt nur noch die Pigtails hinlöten. Bitte aufpassen das der Isolator (Gummi), zwischen Hauptader in der Mitte, und dem Geflecht außen rum noch sauber isoliert.



Und so in der Art solltet ihr des dann machen....



Achtung - Hinweise zum obigen Bild: So in der Art sollte es besser nicht aussehen da ihr evtl. zu hohe Verluste wegen mangelnder Schirmung habt. Ich werde demnächst neue Fotos posten, in denen diese Problem ausgemerzt wurde.

Das RP-SMA-Ende könnt ihr dann wie ich mit Power-Knete befestigen. Ansonsten Heißkleber, dass bleibt aber euch überlassen. Der Rest sollte klar sein, wieder alles schön zusammenbauen und Spaß haben.

Version herausfinden

Die ersten vier Stellen der Seriennummer (zu finden auf der Unterseite des Routers) repräsentieren die Hardware-Version:

```
CDF0 = WRT54G v1.0
CDF1 = WRT54G v1.0
CDF2 = WRT54G v1.1
CDF3 = WRT54G v1.1
CDF5 = WRT54G v2.0
CDF7 = WRT54G v2.2
CDF8 = WRT54G v3.0
CDF9 = WRT54G v3.1
CDFA = WRT54G v4.0
CDFB = WRT54G v5.0
CDFC = WRT54G v5.1
CDFB = WRT54G v5.2
CDFD = WRT54G v6.0
CDFE = WRT54G v7.0 (Atheros AR2317 Chipset) Nicht DD-WRT kompatibel
CDFK = WRT54G v7.2
CDFE = WRT54G v8.0
MDF0 = WRT54G v8.1
```

CDFJ = WRT54G v8.2

CGN0 = WRT54GS v1.0

CGN1 = WRT54GS v1.0

CGN2 = WRT54GS v1.1

CGN3 = WRT54GS v2.0

CGN4 = WRT54GS v2.1

CGN5 = WRT54GS v3.0

CGN6 = WRT54GS v4.0

CGN7 = WRT54GS v5.0

CGN8 = WRT54GS v5.1

CGN9 = WRT54GS v6.0

CGNA = WRT54GS v7.0

CGNB = WRT54GS v7.0

CGNC = WRT54GS v7.0

CGNE = WRT54GS v7.2

CL7A = WRT54GL v1.0

CL7B = WRT54GL v1.1

CL7C = WRT54GL v1.1

CF7C = WRT54GL v1.1

Informationen über Hardware

Infos über die Hardware erhaltet ihr auf folgenden Seiten:

- ◆ <http://Router-IP/SysInfo.htm> (die Versionsangabe hier ist nicht immer zuverlässig - sicherer ist die Seriennummer)
- ◆ <http://Router-IP/Cysaja.asp>
- ◆ <http://Router-IP/Info.htm> (Um diese Seite als Startseite unter <http://Router-IP/> zu sehen, kann man sie im GUI unter "Administration ->Management ->Remote Router Access -> "Enable Status Site" aktivieren.)

NVRAM

NVRAM steht für *non volatile random access memory* (nichtflüchtiger Speicher) und ist sozusagen der Konfigurationsspeicher des WRT. Hier werden sämtliche Parameter und gesetzte Optionen gespeichert. Die Shell stellt dazu den Befehl *nvr* bereit. Mit

```
nvr show
```

kann man sich den gesamten Inhalt auflisten. Mittels *grep* kann man die Ausgabe auf die Variablen einzelner Bereiche eingrenzen und übersichtlicher darstellen, z.B.

```
nvr show | grep jffs2
```

. Mit *nvr get* und *nvr set* kann man Parameter auslesen bzw. ändern. *nvr commit* führt die Änderungen dann aus.

```
erase nvr
```

Mit *erase* kann man NVRAM Löschen

Overclocking der CPU

Mit folgendem Befehl wird die CPU übertaktet (hier auf 240 MHz):

```
nvrans set clkfreq=240
nvrans commit
reboot
```

Um die aktuelle MHz-Einstellung zu erhalten folgenden Befehl eingeben:

```
nvrans get clkfreq
```

oder

```
cat /proc/cpuinfo
```

Übertakten ist insbesondere eine Lösung, wenn es bei Netzwerklast zwischen WLAN<=>LAN plötzlich zu Reboots, WLAN-Disconnects usw. kommt. Betroffen sind besonders die Modelle WRT54G v2.2, v3 und GS v.1, v2.0 bei denen meistens nach ein paar Sekunden, einer halben Stunde oder auch mehreren Stunden bei durchgängiger Last (~30 kB/s) diese Probleme auftreten.

JFFS2

Journaling Dateisystem

Journaled File System (JFS)

Journaling Flash File System (mtd-jffs-HOWTO).

Erweiterungen

USB Support

Installation:

Für die Unterstützung von USB müssen folgenden Pakete installiert werden :

```
ipkg install kmod-usb-uhci // für USB 1.1
ipkg install kmod-usb-core
ipkg install kmod-usb2
ipkg install kmod-usb-storage //für USB-Massenspeicher
```

Bei DD-WRT v23 oder niedriger muss noch eine Kleinigkeit in die NVRAM Variable rc_startup geschrieben werden. Dies führt beim Start alle *.startup Skripte unter /jffs/etc/config/ aus.

In einer Shell auf dem Router:

Erweiterungen

```
> ~ # nvram set rc_startup="
> for I in `bin/ls /jffs/etc/config/*.startup`
> do
>   sh $I &
> done
> "
> ~ # nvram commit
>
```

Über das Webinterface

```
for I in `bin/ls /jffs/etc/config/*.startup`
do
  sh $I &
done
```

Dies ist das Script mit dem die für USB nötigen Kernel Module beim ausführen geladen werden. Einfach den Text ausschneiden und in `/jffs/etc/config/usb.startup` ablegen.

```
#!/bin/sh
insmod /jffs/lib/modules/2.4.30/usbcore.o
insmod /jffs/lib/modules/2.4.30/ehci-hcd.o
insmod /jffs/lib/modules/2.4.30/scsi_mod.o
insmod /jffs/lib/modules/2.4.30/usb-storage.o
# Für USB 1.1 Support die Raute (#) vor der nächsten Zeile entfernen
#insmod /jffs/lib/modules/2.4.30/uhci.o
insmod /jffs/lib/modules/2.4.30/sd_mod.o
```

das Script ausführbar machen

```
chmod +x /jffs/etc/config/usb.startup
```

Auf einem Asus WL-500gPremium sollte es bei eingabe von dem Komando `dmesg` nun so aussehen:

```
SCSI subsystem driver Revision: 1.00
Initializing USB Mass Storage driver...
usb.c: registered new driver usb-storage
USB Mass Storage support registered.
uhci.c: USB Universal Host Controller Interface driver v1.1
PCI: Enabling device 01:03.0 (0000 -> 0001)
uhci.c: USB UHCI at I/O 0x100, IRQ 2
usb.c: new USB bus registered, assigned bus number 2
hub.c: USB hub found
hub.c: 2 ports detected
PCI: Enabling device 01:03.1 (0000 -> 0001)
uhci.c: USB UHCI at I/O 0x120, IRQ 2
usb.c: new USB bus registered, assigned bus number 3
hub.c: USB hub found
hub.c: 2 ports detected
```

Je nachdem, welches Dateisystem auf dem Datenträger eingesetzt wird, müssen noch die Filesystem (FAT/EXT/XFS) Module installiert werden:

```
ipkg install kmod-vfat
ipkg install kmod-ext2
```



```
ipkg install kmod-ext3
ipkg install kmod-xfstools
```

Damit diese Module beim Neustart mitgeladen werden, muss der folgende Text noch an `/jffs/etc/config/usb.startup` angehängt werden.

```
# FAT
insmod /jffs/lib/modules/2.4.30/fat.o
insmod /jffs/lib/modules/2.4.30/vfat.o
# XFS
insmod /jffs/lib/modules/2.4.30/xfstools.o
# EXT
insmod /jffs/lib/modules/2.4.30/jbd.o
insmod /jffs/lib/modules/2.4.30/ext2.o
insmod /jffs/lib/modules/2.4.30/ext3.o
```

Die Partitionen der ersten USB Platte sind nun über `/dev/scsi/host0/bus0/target0/lun0/part1` bis `partN` erreichbar.

Installation eines Samba Servers

Wenn man einen Router mit schneller CPU, genug Speicher und funktionierenden USB 2.0 Ports (wie den ASUS WL-500P) sein eigen nennt, möchte man sicher die vorzüge der Windows Datei Freigabe nutzen. Um dies zu tun muss man Samba installieren.

```
ipkg install samba
```

nach einigem ruminstallieren ist alles fertig für einen ersten Test. Leider bricht `/jffs/etc/init.d/samba` start mit diesem Fehler ab:

```
nmbd: can't resolve symbol 'fopen64'
smbd: can't resolve symbol 'stat64'
```

Durch eine Installation von `uclibc_0.9.28-6_mipsel.ipk` lässt sich dies beheben. Ein nettes Howto hierzu leider in Englisch <http://www.dd-wrt.com/wiki/index.php/Optware>

Zum Abschluss braucht noch die `smb.conf` einige Anpassungen (`/jffs/etc/samba/smb.conf`).

FTP-Server

Lösung 1

von: Helmutoh

Um einen FTP-Server auf dem Router laufen zu lassen sind die folgenden Schritte notwendig:

Zunächst sollte sichergestellt sein, dass genügend Speicherplatz im JFFS-Verzeichnis vorhanden ist, um die notwendigen Pakete zu installieren. Sollte dies nicht der Fall sein (vor allem wenn man auch noch Daten per FTP irgendwo speichern möchte) empfiehlt sich das Einbinden einer SMB-Windows-Freigabe oder gar der Einbau einer SD-Karte. Um diese Erklärung einfach zu halten verwende ich bei allen Pfadangaben hier nur

den zur SD-Karte. Zudem zeige ich die Installation der Zusatzpakete auch auf der SD-Karte.

Vorbereitungen:

Da die Pakete nicht im Standard-JFFS installiert werden benötigen wir die entsprechenden Verzeichnisse (wer im JFFS genügend Platz hat, braucht diesen Schritt natürlich nicht!):

```
mkdir /mmc/jffs
mkdir /mmc/jffs/tmp
mkdir /mmc/jffs/tmp/ipkg
```

Schritt 1:

Installation des vsftpd-Server Packages von downloads.openwrt.org

```
ipkg -d /mmc/jffs install http://URL 2 vsftpd
```

Achtung hier bitte jeweils die URL der aktuellsten Version des vsftpd-Paketes einsetzen! Bei der Installation mit ipkg ggf. die Warnmeldungen ignorieren!

Schritt 2:

Installation der zusätzlichen Library

Da der FTP-Server leider nicht mit den Standard-Libs in DD-WRT startet benötigen wir ein zusätzliches Paket (uclibc):

```
ipkg -d /mmc/jffs install http://URL 2 uclibc
```

Auch hier wieder die Version beachten und richtige URL zum uclibc-Paket einsetzen!

Schritt 3:

Start-Skript für den FTP-Server

Zum Start des FTP-Server gehört nun ein Start-Skript, welches kurzzeitig die Lib aktiviert und den Server damit startet.

```
export LD_PRELOAD='/mmc/jffs/lib/libuClibc-0.9.27.so'
/mmc/jffs/usr/sbin/vsftpd /mmc/jffs/etc/vsftpd.conf
unset LD_PRELOAD
env > /dev/null
```

In der zweiten Zeile wird der FTP-Server mit seiner Config-Datei gestartet. Hier sollte unbedingt festgelegt werden, welcher USER was darf! Die Optionen dieser Config finden sich am besten hier: [vsftpd manpage](#).

Dieses Skript an beliebiger Stelle speichern (hier: /mmc/progs/ftpserver/start).

Schritt 4:

Das erstellte Start-Skript kann nun nach erfolgreichem Test auch im Startup-Bereich des Router eingetragen werden:

Im Bereich Administration - Diagnostics der Router-Konfiguration

```
cd /mmc/progs/ftpserver
./start
```

mit Save Startup abspeichern.

Schritt 5:

Soll der FTP-Server nun auch noch im WAN (Internet) sichtbar sein, so muss der Port 21 für passive FTP-Verbindungen noch in der Firewall geöffnet werden:

Im Bereich Administration - Diagnostics

```
/usr/sbin/iptables -I INPUT 1 -p tcp --dport 21 -j logaccept
```

mit Save Firewall abspeichern.

Lösung 2 von: [moscito](#)

vorweg erstmal ein dickes DANKE für die infos aus dem forum.

mein gerät ist ein (WRT54GL) mit DD-WRT v23_sp2 und sp3, habe es mit beiden versionen gemacht. ich selber habe mit der Lösung 1 nichts erreicht.

Vorbereitung: MMC in das /jffs mounten

```
mount -o bind /mmc /jffs
mount -o bind /mmcopt/opt /opt
```

Pakete installieren: erstmal den Paketmanager "frisch" machen.

```
ipkg update
ipkg install vsftpd
ipkg install uclibc
```

das startscript erstellen

```
cd /mmc
```

mit *touch ftpstartup* wir die datei erzeugt mit *vi ftpstartup* editiert man die datei

die taste i bringt dich in den schreibmodus mit ESC wir der beendet und dann gehts mit :wq wieder raus. verlassen ohne zu speichern geht mit :q bzw :q!

Einfach die sachen zwischen den " " mit vi eintippen. zweite möglichkeit wäre ohne vi auf der console

```
echo "sleep 60" >> /mmc/ftpstartup
echo "mount -o bind /mmc /jffs" >> /mmc/ftpstartup
echo "mount -o bind /mmcopt/opt /opt" >> /mmc/ftpstartup
echo "echo "nobody:Passwort:0:0:User,,,:/mmc:/bin/sh" >> /tmp/etc/passwd" >> /mmc/ftpstartup
echo "export LD_PRELOAD='/mmc/jffs/lib/libuclibc-0.9.27.so'" >> /mmc/ftpstartup
echo "/mmc/jffs/usr/sbin/vsftpd /mmc/jffs/etc/vsftpd.conf" >> /mmc/ftpstartup
echo "unset LD_PRELOAD" >> /mmc/ftpstartup
```

```
echo "env > /dev/null" >> /mmc/ftpstartup
```

Nun starte den ftp server mit

```
./ftpstartup
```

wenn das script automatisch starten soll dann in der GUI des DD-WRT "Administration"----"Diagnose" aufsuchen und die folgenden 2 zeilen eintragen und speichern als startup

```
cd /mmc  
./ftpstartup
```

mit ps sollte vsftp in der prozessliste zu sehen sein.

soll der ftp aus dem www erreichbar sein dann bitte den schritt 5 dieser wiki nutzen. drann denken das es nicht ganz ungefährlich ist den root für ftp frei zu geben. also bitte immer fein die backups machen.....vorher :)

RFlow Collector

Mit diesem Programm kann man die Aktivität des Routers überwachen. Es gibt Auskunft über verbundene Clients und zeigt auch die Empfangsstärke an.

Trouble Shooting

Der WRT54-g(s)(L) macht kaum Probleme

Ich möchte mitmachen

Ganz einfach: Den "edit"-Button drücken, editieren, speichern und uns erfreuen. ;)

Alphabetischer Index

Hier sind alle projektbezogenen Artikel alphabetisch gelistet. **DD-WRT Doku(DE) Index**

Anpassen der Ports des Switches auf Halbduplex und Einstellung von Geschwindigkeiten.