

Pimp my router (Siemens SE 505)

Jan Helber

This documentation describes the modification of a Siemens Gigaset SE 505 router: An USB connector and the power supply for this connector is soldered to the circuit layer of the router. As operating system OpenWRT-Linux is being installed.

Another router (ASUS) is setup. On both routers a 1TB USB-storage is setup to have a decentral RAID1. The goal is to have about 500GB of mirrored disk space in two households. Therefore an incremental backup is run automatically each night: Half of the disk space is mirrored from Siemens to ASUS and the the other half is mirrored the other way round.

Powered by \LaTeX

Generated on July 18, 2010 at 18:55 o'clock

Contents

1	Introduction	1
2	USB-Buchse einlöten	2
3	RAM aufrüsten	4
3.1	Info	4
3.2	Welche RAM Bauform?!	4
3.3	Anleitung um auf 16MB aufzurüsten (Viel hilft viel)	4
3.4	Tipps	5
3.5	Quellen für RAM-Chips	5
4	Compile OpenWrt	5
4.1	Loop AES	6
5	Linux installieren & einrichten	7
5.1	TFTP - Router flashen	8
5.2	Minimal system setup	8
5.2.1	Wireless LAN	9
5.2.2	Network configuration	9

5.2.3	Static IP (MAC based)	11
5.2.4	Firewall	11
5.3	Packete installieren	12
5.3.1	USB 1.1 & USB 2.0 support for mass storage devices	13
5.3.2	Partition and format USB-storage	14
5.3.3	Mount USB-storage	18
5.3.4	Aktiviere swap partition	18
5.3.5	Installation von Paketen auf USB-storage vorbereiten	18
5.3.6	Loop AES	19
5.3.7	Samba	21
5.3.8	Performance	23
5.3.9	Cron	23
5.3.10	DynDns	25
5.3.11	Rsync	25
5.3.12	Zeitzone	26
5.3.13	RDate	27
5.3.14	OpenVPN	27
5.3.15	Socks5	29
5.3.16	WOL	29
5.3.17	QoS	30
5.4	Shell-Skripte	31
5.5	MAC address cloning	34
5.6	SSH-Public-Key-Authentification	34
6	Netzwerk Konfiguration	35
6.1	WiFi	35
6.2	LAN	37
7	Backup	38
8	Information about Siemens SE 505	38
9	Viel Spaß!	38
10	FAQ	38

11 Anhang	39
12 Quellen	39
13 Marken	40
14 Haftungsausschluss	40

List of Figures

1 Router	3
2 Casing (backside)	1
3 Platine (Vorderseite)	2
4 Platine (Rückseite)	2
5 Platine (nach dem Löten)	3
6 Gehäuse (angelötete USB-Buchse)	41
7 Gehäuse (von unten)	41

List of Tables

1 WiFi NVRAM	36
2 LAN NVRAM	37

Figure 1: Router



1 Introduction

The status of this howto is work in progress, cause it is continuously updated and translated into English step by step. Perhaps you might want to have a look on my more recent OpenWrt setup: <https://www.helber.info/fileadmin/Projekte/Router/OpenWrt-Router-ASUS-W.pdf>

I bought my self a Siemens Gigaset SE 505 router at eBay (including transportation you shouldn't pay more than 35 Euro). You should make sure, that you buy version 2, cause the pinning to solder in the USB-connector is unknown for version 1.

At first view both router-variants look the same. However the internal circuit layer differs and therefore the alignment of the connectors - visible from the outside - differs a bit as well.

Easiest to discriminate is the position of the reset pushbutton:

- At version 1 it is located in the middle between WAN connector and antenna.
- As you can see in figure 2 at version 2 it is located a bit closer to the WAN connector.

Figure 2: Casing (backside)



Another identifying feature is the serial-number. If you can find "S30853-S1006-R107-3 S8" on the backside of the packaging, everything should be fine.

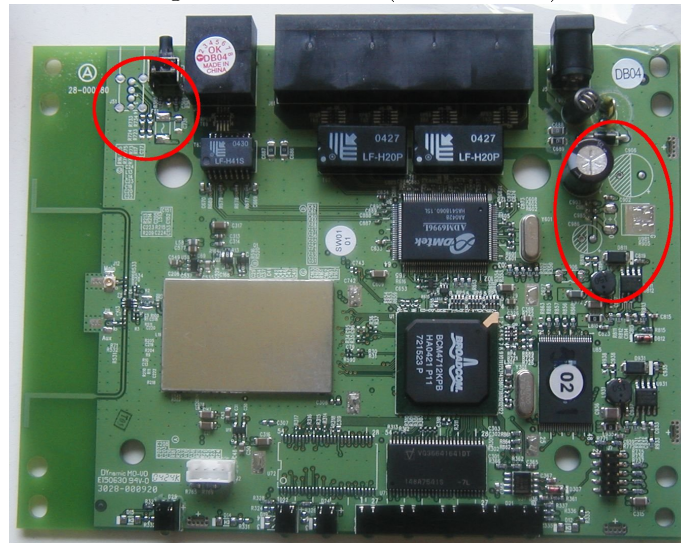
Rumour has it that you can identify version 1 by a power connector labeled with AC (alternating current), at version 2 it would be labeled with DC (direct durrent). This is WRONG! There are definitely devices of version 2 labeled with AC as well.

At times in this documentation there are commands that require some linux-knowledge (e.g. the setup-scripts). I added them, cause I don't want to forget them during a reinstallation. This is kind of a howto for myself as well (makes live easier *g*).

2 USB-Buchse einlöten

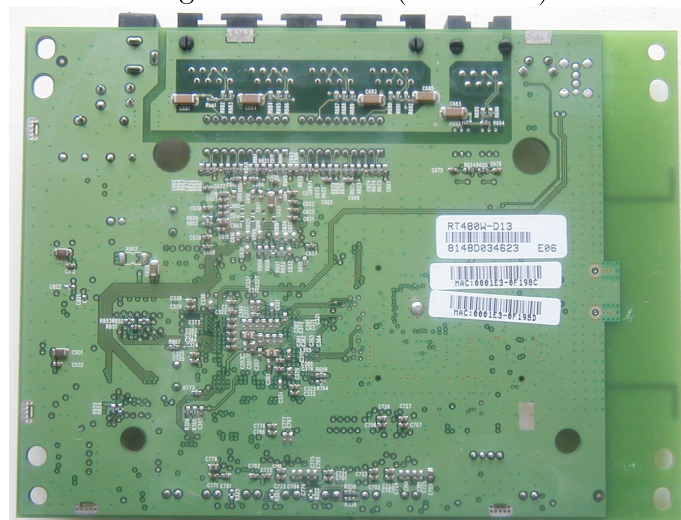
Zunächst öffnet man das Gehäuse indem man die 4 Schrauben auf der Rückseite abschraubt und das Antennenkabel von J12 abzieht dann dürfte das ganze so oder so ähnlich aussehen:

Figure 3: Platine (Vorderseite)



Die Stellen, an denen die Bauteile hin gelötet werden müssen habe ich rot umrandet.

Figure 4: Platine (Rückseite)



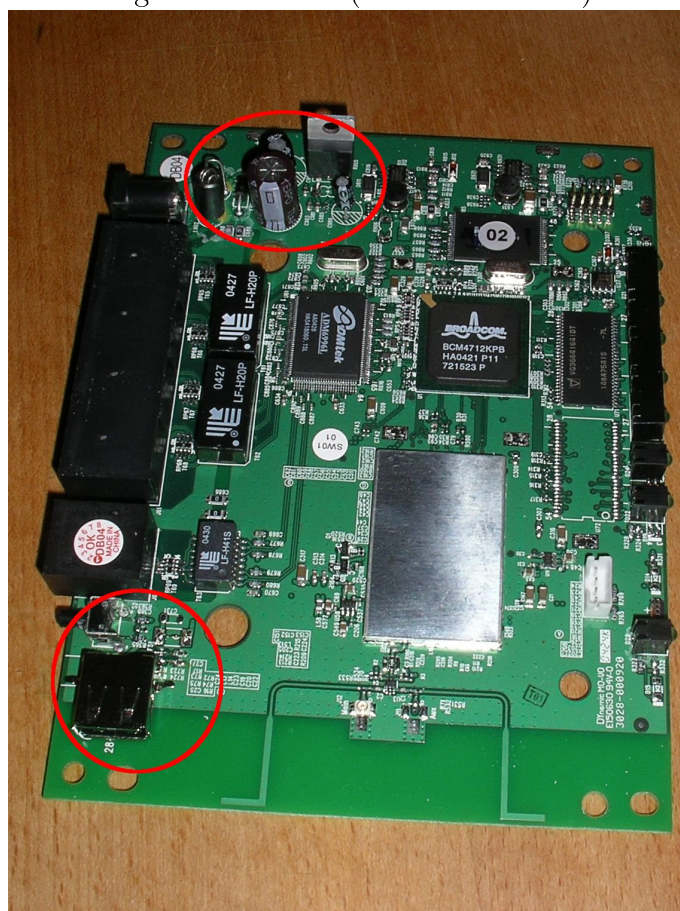
Zum Löten sollte man einen möglichst feinen LötKolben besitzen, der für SMD-Bauteilen geeignet ist. Besonders das Löten der kleinen Widerstände ist nicht ganz trivial.

Auf der Seite der Spannungsversorgungs-Buchse müssen folgende Bauteile aufgelötet werden:

2 USB-BUCHSE EINLÖTEN

- C906 : $100\mu F$ 16Volt (< 20Cent)
- C986 : $10\mu F$ 16Volt (< 15Cent)
- U981 : LM7805 (Spannungsregler) (< 60Cent)
- F51: Draht als Sicherung drauf löten :)
- R723: 15kOhm (<10Cent)
- R724: 15kOhm
- R733: mit Lötzinn überbrücken
- R734: mit Lötzinn überbrücken
- J51: USB-Buchse Typ A (<1,50Euro)

Figure 5: Platine (nach dem Löten)



Da man bei geschlossenem Gehäuse nicht mehr an die USB-Buchse dran kommt muss man das Gehäuse neben der WLAN-Antenne etwas aufzwickeln. Dabei muss ein Steg (zum fest schrauben der Rückseite) entfernt werden (siehe Abbildung 6). Aber die übrigen 3 reichen

völlig.

Ich habe den Spannungsregler senkrecht (siehe Abbildung 5) auf die Platine gelötet, um noch einen Kühlkörper anbringen zu können. Dazu mischt man am besten etwas Sekundenkleber und Wärmeleitpaste zusammen und befestigt den Kühlkörper auf dem Blechteil des Spannungsreglers. Da der Spannungsregler etwas zu hoch war, hab ich ein kleines Loch ins Gehäuse gebohrt wo er etwas heraus schaut.

3 RAM aufrüsten

3.1 Info

Die popeligen 8MB RAM, die der Router von Haus aus mit sich bringt reichen ja nicht mal um einen Samba-Daemon vernünftig betreiben zu können. Daher empfehle ich den Speicher des Router zu erweitern. Allerdings möchte ich darauf hinweisen, dass dadurch der Garantieanspruch erlischt!

3.2 Welche RAM Bauform?!

Der SE505V2 enthält einen 4Mx16 SDRAM Chip (8 MB), und freie Pads für einen weiteren RAM Chip. Die RAM-Init-Routine des standard CFE¹ ignoriert den Inhalt der `s dram_init` Variable im `nvr am`, wodurch 8Mx16 Chips wie 4Mx16 Chips behandelt werden. Gleichzeitiges Verwenden von 4Mx16 und 8Mx16 ist möglich, wird aber das selbe Ergebnis wie zwei 4Mx16 Chips liefern.

3.3 Anleitung um auf 16MB aufzurüsten (Viel hilft viel)

- 16 Bit SDRAM Chips besorgen (Datenblatt lesen!). 64MB und 128MB SODIMMS sind eine gute Quelle für SDRAM mit der richtigen Busbreite. Das ablöten sollte mit einer Heissluftpistole gut funktionieren (Chips nicht überhitzen!).
- Nun müssen die LEDs vorsichtig abgelötet werden (mindestens die WLAN- und WWW-LED)
- Säubere die RAM-Pads unter Verwendung eines LötKolben mit mindestens 50W und Entlötlitze.
- Löte einen 4Mx16 SDRAM Chip auf die unbestückten RAM-Pads. Darauf achten, dass keine Kurzschlüsse entstehen!
- Um die Adressleitungen für den zweiten RAM-Chip anzuschließen müssen folgenden Pads mit einem SEHR dünnen Draht verbunden werden:

¹CFE steht für Common Firmware Environment

- R314
- R315
- R316
- R317
- R318

Nach einem Reboot sollten nun 16MB RAM zur Verfügung stehen.

Zu guter Letzt sollten folgende Befehle eingegeben werden damit der CFE das zeitliche Signalverhalten rekonfiguriert:

```
nvrnm set sdram_ncdl=0
nvrnm commit
reboot
```

Das wars!

3.4 Tipps

- Leinen Kondensator bei C319 anlöten! Dieser Kondensator ist nur zum Feintuning des RAM vorhanden, für einen stabilen Betrieb des Router aber nicht notwendig.
- Um auf 32MB aufzurüsten werden 16x16MB Chips von einem 4 Chip 128MB, oder einem 8 Chip 256MB SODIMM benötigt. Vorsicht beim ablöten des alten RAM-Chips vom Router!

3.5 Quellen für RAM-Chips

- 4x16MB Durch ablöten der Chips von einem SE505V2 Router ;-)
- 8x16MB 64MB SDRAM PC133 SODIMMs w/ 4 Chips

Die sicherste Methode ist es Chips von einem PC133 SDRAM Modul mit 4 Bausteinen zu verwenden. RAM von manchen SODIMMs oder DIMMs mit 8 Bausteinen und der doppelten Kapazität wie oben erwähnt könnten auch die richtigen Eigenschaften haben. Es sollte vorher immer das Datenblatt gelesen werden!

4 Compile OpenWrt

If you don't want to compile OpenWrt by your own, just skip this step and continue at section 5 on page 7.

First of all you need to download the latest version of OpenWrt via SVN somewhere to your Linux PC. In my case it was revision 14413 and I downloaded it to ~/Openwrt/trunk:

```
mkdir ~/Openwrt
cd ~/Openwrt
svn co -r 14413 https://svn.openwrt.org/openwrt/trunk
```

Next the prerequisites need to be checked. The following command only informs about missing programs, but they need to be installed manually:

```
make prereq
```

When all missing programs on the desktop PC were installed, the feed packages should be updated:

```
scripts/feeds update
make package/symlinks
```

With the following command you can configure your binary which will be flashed to the server (marked with *****) and also the packages available as module for later installation (marked with **M**):

```
make menuconfig
```

Depending on the packages that you selected for your OpenWrt build some more prerequisites might be necessary. Therefore the prerequisites should be rechecked before installation:

```
make prereq
make world V=99
```

4.1 Loop AES

Before we install the loop-aes, some comments on building OpenWrt from SVN. The latest official release of OpenWrt Kamikaze 7.09 doesn't include loop-aes. Fortunately, the current SVN version DOES include a patched loop kernel module ("kmod-loop-aes"), but no patched userland tools. So, with the SVN version we have to go only half the way to an encrypted system by patching the userland tools. [[Filesystem Encryption on OpenWrt](#)]

First check in ~/Openwrt/trunk/package/util-linux-ng/Makefile which version of util-linux-ng is included in the svn (mine was `PKG_VERSION:=2.13.0.1`).

Then download the loop-aes version, which corresponds to that version (mine: loop-AES-v3.2b):

```
cd ~/Openwrt/
wget http://loop-aes.sourceforge.net/loop-AES/loop-AES-v3.2b.tar.bz2
tar -jxvf loop-AES-v3.2b.tar.bz2
```

There's a file included: util-linux-ng-XXX.diff (in my case util-linux-ng-2.13.0.1.diff). Copy it to trunk/package/util-linux-ng/patches and rename it to something like 002-util-linux-ng-2.13.0.1.diff depending on how many patches there are already:

```
cd ~/Openwrt/loop-AES-v3.2b
#make
cp util-linux-ng-2.13.0.1.diff \
../trunk/package/util-linux-ng/patches/002-util-linux-ng-2.13.0.1.diff
```

Next, edit trunk/package/util-linux-ng/Makefile and replace all occurrences of “losetup”, “mount-utils” and “swap-utils” with “aeslosetup”, “aesmount-utils”, “aesswap-utils” (not in Build/Compile of course). We want to avoid conflicts with the original packages and distinguish them better from the original packages. (In fact, it is possible to clone the whole package and build a patched and unpatched version simultaneously.):

```
vi ../trunk/package/util-linux-ng/Makefile
#:%s/mount-utils/aesmount-utils/gi
#:1,92s/losetup/aeslosetup/gi
#:97,$s/losetup/aeslosetup/gi
#:%s/swap-utils/aesswap-utils/gi
```

Since aeslosetup doesn't exist till now we simply create a symlink. Otherwise the build-process would return with an error:

```
cd ~/Openwrt/trunk
ln -s ~/Openwrt/trunk/build_dir/target-mipsel_uClibc-0.9.29/util-linux-ng-2.13.0.1/mo
~/Openwrt/trunk/build_dir/target-mipsel_uClibc-0.9.29/util-linux-ng-2.13.0.1/mount/ae
ls build_dir/target-mipsel_uClibc-0.9.29/util-linux-ng-2.13.0.1/mount/ \
| grep losetup
```

At last:

```
cd ~/Openwrt/trunk
make menuconfig
#Utilities->e2fsprogs, Utilities->disc->aesswap-utils
make world V=99
```

For being able to install packages on the router from the linux PC:

```
chmod o+r ~/Openwrt/trunk -R
ln -s ~/Openwrt/trunk /var/www/trunk
```

5 Linux installieren & einrichten

Since the USB connector is not supported by the original firmware installed on the router we need another firmware. Since OpenWrt is very stable (hell meets our needs best).

Damit die USB-Buchse ihren Dienst verrichten kann muss als nächstes das openWRT² (Linux) installiert werden. Dazu habe ich mir die OpenWrt Kamikaze 8.09 RC1 brcm-2.4 (openwrt-brcm-2.4-squashfs.trx) von folgender Adresse herunter geladen:

http://downloads.openwrt.org/kamikaze/8.09_RC1/brcm-2.4/

Damit diese Anleitung keine Probleme mit verschiedenen Versionen bekommt habe ich die von mir verwendete Version auf meinem Webspaces zum Download bereit gestellt:

<http://www.helber.it/fileadmin/Projekte/Router/openwrt-brcm-2.4-squashfs.trx>

5.1 TFTP - Router flashen

Praktischerweise ist unter Windows XP und Linux schon standardmäßig ein TFTP-Client vorinstalliert. Um den Router mit der neue Firmware flashen zu können, muss der Computer und der Router mit einem normalen Netzkabel (kein Crossover) am WAN Port (beim Siemens LAN4) verbunden sein. Die IP-Adresse des Computers (192.168.1.100) muss statisch in der Konfiguration eingetragen werden. Nun resettet man den Router, indem man den Stromstecker zieht und (z.B. mit einer Büroklammer) und den RESET-Knopf gedrückt hält während man die Stromversorgung wieder herstellt. Sobald beim Siemens die 5 Front-LEDs (LAN1-4 & WAN) aufleuchten (beim ASUS fängt die Powerleuchte an langsam zu blinken) führt man folgenden Befehl von der Konsole des Computers (Windows: cmd) aus:

Windows:

```
tftp -i 192.168.1.1 PUT openwrt-brcm-2.4-squashfs.trx
```

Linux:

```
cd /home/jan/Openwrt/trunk/bin
```

```
atftp --trace -p -l openwrt-brcm-2.4-squashfs.trx --verbose 192.168.1.1
```

Wenn alles funktioniert hat erscheint danach etwas ähnliches wie:

Übertragung erfolgreich: 1970176 Bytes in 12 Sekundens, 164181 Bytes/s

Nun ist etwas geduld gefragt. Die WAN-LED blinkt zunächst sehr hektisch (er schreibt das neue Image auf den Flash). Danach sollte der Router innerhalb von wenigen Minuten automatisch Neustarten. Das erkennt man daran, dass alle 5 LAN-LEDs mehrfach blinken. Sollte er sich auch nach 5 Minuten nicht anpingen lassen (192.168.1.1), dann sollte ein normaler Reset mit der Büroklammer weiterhelfen.

5.2 Minimal system setup

Afterwards you log in via telnet on the router. Therefore no password is needed. The first thing that should be done is to set a new password with the command "passwd". This

²<http://wiki.openwrt.org>

will automatically deactivate the telnet daemon and activate the SSH daemon (dropbear).

For questions about problems with your router it is often necessary to know which version you are using. You can figure this out by the following command:

```
uname -a

# In my case it was:
# Linux OpenWrt 2.4.35.4 #41 Tue Nov 4 01:20:52 UTC 2008 mips unknown
```

5.2.1 Wireless LAN

To set up the WLAN with WPA³ encryption, the file `/etc/config/wireless` needs to be edited. An example how this is to be done is listed below. Of course you have to replace "12345" with your own password.

```
echo "config wifi-device wl0
      option type      broadcom
      option channel   5

      # REMOVE THIS LINE TO ENABLE WIFI:
      option disabled 0

config wifi-iface
      option device     wl0
      option network    lan
      option mode       ap
      option ssid       OpenWrt
      option encryption psk
      option key         12345" > /etc/config/wireless

# Load the new settings
wifi down
wifi up
iwconfig
```

5.2.2 Network configuration

Two separate subnets
(e.g. router direct connected with Internet):

```
echo "#### VLAN configuration
```

³wpa2 might cause trouble with XP SP2 and cheap WLAN devices

```
config switch eth0
    option vlan0    \"1 2 3 4 5*\
    option vlan1    \"0 5\

#### Loopback configuration
config interface loopback
    option ifname    \"lo\
    option proto     static
    option ipaddr    127.0.0.1
    option netmask   255.0.0.0

#### LAN configuration
config interface lan
    option type      bridge
    option ifname    \"eth0.0\
    option proto     static
    option ipaddr    192.168.1.1
    option netmask   255.255.255.0

#### WAN configuration
config interface    wan
    option ifname    \"eth0.1\
    option proto     dhcp" > /etc/config/network
```

Switched network (behind another router):

```
echo "#### VLAN configuration
config switch eth0
    option vlan0    \"1 2 3 4 5*\
    option vlan1    \"0 5\

#### Loopback configuration
config interface loopback
    option ifname    \"lo\
    option proto     static
    option ipaddr    127.0.0.1
    option netmask   255.0.0.0

#### LAN configuration
config interface lan
    option type      bridge
    option ifname    'eth0.0'
```

```

        option proto      static
        option ipaddr     192.168.1.112
        option netmask    255.255.255.0
        option 'gateway'  '192.168.1.1'
        option 'dns'      '192.168.1.1'

#### WAN configuration
#config interface      wan
#      option ifname   \"eth0.1\"
#      option proto     dhcp" > /etc/config/network

```

5.2.3 Static IP (MAC based)

<http://johnbokma.com/mexit/2008/09/03/dhcp-static-ip-dnsmasq.html> To make it easier to reach each device in the network, each device has its own IP-address. An example how this is to be done is listed below:

```

echo "# Laptop1
00:18:de:e2:08:61 192.168.1.200
# Serverhome
00:30:05:8a:51:ff 192.168.1.240
# Laptop2
00:13:49:70:54:81 192.168.1.242
# Siemens
00:90:96:00:00:00 192.168.1.142" > /etc/ethers
# First we need to delete the symlink to /var/etc/hosts
rm /etc/hosts
# We want /etc/hosts to be persistent
echo "127.0.0.1 localhost. OpenWrt
192.168.1.1  asus
192.168.1.142 siemens
192.168.1.200 laptop
192.168.1.240 serverhome
192.168.1.242 laptop2" > /etc/hosts

```

5.2.4 Firewall

We want to allow SSH connections from outside the lan (wan):

```

echo "
# Accept SSH-port from wan
config rule
        option src          wan
        option dest_port    22
        option target       ACCEPT

```

```

        option proto          tcp

# Accept HTTPS-port from wan
config rule
    option src                wan
    option dest_port          443
    option target              ACCEPT
    option proto              tcp

# Redirect HTTPS-port requests from WAN to SSH-port
config redirect
    option src                wan
    option src_dport          443
    option dest                lan
    option dest_ip            192.168.1.1
    option dest_port          22
    option proto              tcp" >> /etc/config/firewall

/etc/init.d/firewall stop
/etc/init.d/firewall start

```

Customized firewall settings:

```
vi /etc/config/firewall
```

Somewhere in the file you will find something similar to this:

```

# include a file with users custom iptables rules
#config include
#     option path /etc/firewall.user

```

Remove the leading hash in front of config and option.

5.3 Pakete installieren

If you compiled all necessary packages by your own (section 4 on page 5), you should insert the following as first line in `/etc/opkg.conf`:

```
src/gz snapshot http://<IP>/trunk/bin/packages/target-mipsel_uClibc-0.9.29/
```

The IP of my Linux PC is 192.168.1.240, therefore my `/etc/opkg.conf` looks like this:

```

echo -n "src/gz snapshot " > /etc/opkg.conf
echo "http://192.168.1.240/trunk/bin/packages/target-mipsel_uClibc-0.9.29/
#src/gz snapshots http://downloads.openwrt.org/kamikaze/8.09/brcm-2.4/packages

```

```
##src/gz snapshots http://downloads.openwrt.org/snapshots/brcm-2.4/packages
dest root /
dest ram /tmp
lists_dir ext /var/opkg-lists
option overlay_root /jffs" > /etc/opkg.conf
```

You can also use my package repository. In that case insert the following as first line in /etc/opkg.conf:

```
#ToDo JHR (This repository is not yet up and running!!!)
src/gz snapshot http://helber.it/trunk/bin/packages/target-mipsel_uClibc-0.9.29/
```

Mit folgenden Befehlen kann man z.B. das Paket tcpdump installieren:

```
# Paketliste updaten (geht nur, wenn der Router eine Netzwerkverbindung hat)
opkg update

# tcpdump downloaden und installieren
opkg install tcpdump
```

Das Installieren von Paketen braucht auf dem Router (je nach Paket) viel Ressourcen und kann sehr lange dauern⁴. Zur Not resettet man den Router einfach durch drücken auf den Resetknopf. Wirklich viel kann nicht passieren, weil bei einer Neuinstallation durch ipkg die fehlenden Konfigurationsdateien noch kopiert werden.

Wenn man irgendwann mal folgende Fehlermeldung bekommt "The following package were previously requested but have not been installed", und man möchte die Pakete doch nicht mehr installieren, weil man weiss, dass es sowieso keinen Sinn mehr macht, dann muss man die Datei "/usr/lib/ipkg/status" editieren. Dabei löscht man alle Pakete aus der Liste bei denen steht "install ok not-installed".

5.3.1 USB 1.1 & USB 2.0 support for mass storage devices

With the following commands on the console of both routers the necessary packages for USB storage devices are being installed:

```
# Without this command you might use old package-versions
opkg update
# USB common
opkg install kmod-usb-core
opkg install kmod-usb-ohci
# USB storage
```

⁴Und wenn ich sage lange ... dann meine ich das auch ... teilweise passiert in der Konsole 1-2 Stunden nix ... so lange die SSH-Verbindung nicht abbricht wartet man am besten

```
opkg install kmod-scsi-core
opkg install kmod-usb-storage
# EXT3 and FAT32
opkg install kmod-fs-ext3
opkg install kmod-fs-vfat
# To enable lsusb
#opkg install usbutils
reboot
```

With the following commands on the console of the Asus router the necessary packages for USB 2.0 are being installed:

```
# USB 2.0 for my ASUS
opkg install kmod-usb2
opkg install kmod-nls-utf8 kmod-nls-iso8859-1
reboot
```

5.3.2 Partition and format USB-storage

Precondition: Section [5.3.1](#) on page [13](#).

To be able to partition the storage-device, fdisk needs to be installed:

```
opkg install fdisk
```

The idea of the following partitioning structure is the following:

- Part 1:
 - Filesystem: Ext3
 - Size: ca 91MB
 - Use: More space to install programs (internal mem is to less)
- Part 2:
 - Filesystem: Swap memory
 - Size: ca 200MB
 - Use: Especially for rsync of big partitions (on my setup 2x 460GB)
- Part 3:
 - Filesystem: Ext2 (AES encrypted)⁵

⁵Some might wonder why ext2 is used instead of ext3. This is due to the fact that ext3 is a journaling filesystem. Since the partitions are encrypted ext3 would detect that something is wrong and would destroy the filesystem while trying to repair it.

- Size: $\frac{1}{2}$ of rest (on my setup 460GB)
- Part 4:
 - Filesystem: Ext2 (AES encrypted)
 - Size: $\frac{1}{2}$ of rest (on my setup 460GB)

If an USB-storage is plugged in connector on the router, it is possible to check if the hard disk was recognized with the following command:

```
dmesg | grep USB
```

Something similar to this should show up:

```
hub.c: new USB device 01:03.2-1, assigned address 2
scsi0 : SCSI emulation for USB Mass Storage devices
  Vendor: WDC WD10 Model: EAVS-00D7B1 Rev:
  Type:   Direct-Access ANSI SCSI revision: 02
Attached scsi disk sda at scsi0, channel 0, id 0, lun 0
SCSI device sda: 1953525168 512-byte hdwr sectors (1000205 MB)
Partition check:
 /dev/scsi/host0/bus0/target0/lun0: p1 p2 p3 p4
WARNING: USB Mass Storage data integrity not assured
USB Mass Storage device found at 2
```

First of all we need to know the device name of the USB-stick on the system:

```
# This command lists all partitions of all harddisks:
fdisk -l
```

I partitioned and formatted the device on my linux desktop (much faster than the router):

```
root@helber:~# fdisk -l
```

```
Disk /dev/sda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x9f71e00f
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	19272	154802308+	83	Linux
/dev/sda2		19273	19457	1486012+	5	Extended
/dev/sda5		19273	19457	1485981	82	Linux swap / Solaris

```
Disk /dev/sdb: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x539fda0b
```

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            1       121601    976760001    c   W95 FAT32 (LBA)
root@helber:~#
```

On my setup the devicename of my 1TB USB-storage was `/dev/sdb`. On the router the devicename is usually something similar to `/dev/scsi/host0/bus0/target0/lun0/disc`.

To be able to partition the device all mounted partitions of this device need to be unmounted. This is not necessary on a router, since there is usually no automount. But if you do the partitioning on a linux desktop, your system might mount the device automatically.

```
umount /dev/sdb1
```

Let's partition the USB-storage

```
# On my linux desktop:
fdisk /dev/sdb
```

```
# On the router it would probably be something similar to this:
fdisk /dev/scsi/host0/bus0/target0/lun0/disc
```

```
# Partition; delete
p d
```

```
# new; primary partition; Partition-Nr: 1;
# First cylinder: 1; Last cylinder: 12
```

```
n p 1 1 12
# new; primary partition; Partition-Nr: 2;
# First cylinder: 13; Last cylinder: 37
n p 2 13 37
```

```
# Set partition to Swap partition
t 2 L 82
```

```
# new; primary partition; Partition-Nr: 3;
# First cylinder: 26; Last cylinder: 60813
n p 3 26 60813
```

```
# new; primary partition; Partition-Nr: 4;
# First cylinder: 60814; Last cylinder: 121601
n p 4 60814 121601
```

```
# print the partition table
p
# Write changes to disk!
w
```

The p-command in fdisk should print something similar:

```
Disk /dev/sdb: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xb73a3211
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	12	96358+	83	Linux
/dev/sdb2		13	37	200812+	82	Linux swap / Solaris
/dev/sdb3		38	60819	488231415	83	Linux
/dev/sdb4		60820	121601	488231415	83	Linux

After the USB-storage was partitioned, those partitions need to be formatted with the ext3 filesystem. This can be done with the following commands:

```
# Now the USB-stick is beeing formatted:
mkfs.ext3 /dev/sdb1
```

On the output of the console something similar to this should pop up:

```
mke2fs 1.40.2 (12-Jul-2007)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
244800 inodes, 489468 blocks
24473 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=503316480
15 block groups
32768 blocks per group, 32768 fragments per group
16320 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
```

```
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 20 mounts or 180 days, whichever comes first. Use tune2fs -c or -i to override.

I recommend ext3 on the router USB-stick also for users, that are using windows for their desktop-pc, cause ext3 has the advantage it is a journaling filesystem and therefore the data is much more secure than with using FAT32. Independent of the filesystem Linux, Windows and Mac can access samba-shares on the router.

5.3.3 Mount USB-storage

Following the USB-storage can be mounted:

```
mkdir /mnt/part1
mount -t ext3 /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/part1
cd /mnt/part1
ls -la
```

For further information please refer to: [\[USB storage howto\]](#)

There is lots what you can do with an USB connector on a router. e.g. USB-to-RS232, USB webcam, USB ethernet, USB bluetooth, USB VGA, USB sound ...

Have a look at: <http://wiki.openwrt.org/OpenWrtDocs/Customizing/Hardware/USB>

```
echo "
config mount
    option target    /mnt/part1
    option device    /dev/scsi/host0/bus0/target0/lun0/part1
    option fstype    ext3
    option options   rw, sync
    option enabled   1" >> /etc/config/fstab
```

5.3.4 Aktiviere swap partition

Precondition: Section [5.3.2](#) on page [14](#).

```
mkswap /dev/scsi/host0/bus0/target0/lun0/part2
swapon /dev/scsi/host0/bus0/target0/lun0/part2
free
```

5.3.5 Installation von Paketen auf USB-storage vorbereiten

Precondition: Section [5.3.3](#) on page [18](#).

Mit folgenden Befehlen wird auf dem USB-Stick ein Bereich eingerichtet auf dem Router-Dienste installiert werden können. <http://wiki.openwrt.org/PackagesOnExternalMediaHowTo>

```
# Ordner anlegen:
mkdir /mnt/part1/jffs_export
```

```
# USB-Stick in OPKG-Konfiguration eintragen:
echo "dest usb /mnt/part1/jffs_export" >> /etc/opkg.conf

# USB-Stick beim Hochstarten automatisch mounten:
echo 'mount /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/part1' >> /etc/init.d/extern
echo 'swapon /dev/scsi/host0/bus0/target0/lun0/part2' >> /etc/init.d/externalmount
chmod +x /etc/init.d/externalmount
ln -s /etc/init.d/externalmount /etc/rc.d/S98externalmount

echo "
config mount
    option target    /mnt/part1
    option device    /dev/scsi/host0/bus0/target0/lun0/part1
    option fstype    ext3
    option options   rw,sync
    option enabled   1" >> /etc/config/fstab

mkdir /mnt/part1/bin
cd /mnt/part1/bin
wget http://www.helber.it/fileadmin/Projekte/Router/opkg-link
chmod +x opkg-link
mv opkg-link opkg-link.sh
ln -s /mnt/part1/bin/opkg-link.sh /usr/bin/opkg-link.sh
```

Da der Router nur begrenzt Flashspeicher zur Verfügung stellt, ist dies eine gute Möglichkeit um den Router um jede Menge Funktionalität erweitern zu können.

5.3.6 Loop AES

Precondition: Section [5.3.2](#) on page [14](#).

Let's install all the necessary packages. Usually those packages are installed on the root of the filesystem, since we can't install packages for USB-access on an USB storage device ... *g*

```
opkg update
# Should be installed on USB-fs:
opkg install -dest usb kmod-crypto-core
opkg install -dest usb kmod-crypto-aes
opkg install -dest usb kmod-loop-aes
opkg-link.sh add kmod-crypto-core
opkg-link.sh add kmod-crypto-aes
opkg-link.sh add kmod-loop-aes
opkg install -dest usb libext2fs
opkg install -dest usb e2fsprogs
opkg install -dest usb aeslosetup
opkg-link.sh add aeslosetup
```

```
opkg-link.sh add e2fsprogs
opkg-link.sh add libblkid
opkg-link.sh add libext2fs
opkg-link.sh add libuuid
opkg install -dest usb aesswap-utils
opkg-link.sh add aesswap-utils
#For the encrypted partitions necessary
opkg install kmod-fs-ext2

# Should be installed on root:
opkg install -force-overwrite aesmount-utils
reboot
```

Some good links:

<http://archiv.raid-rush.ws/t-19217.html>

http://www.pro-linux.de/t_system/loop-aes.html ToDo JHR

```
mkfs.ext3 /dev/scsi/host0/bus0/target0/lun0/part1
aeslosetup -e AES256 /dev/loop/0 /dev/scsi/host0/bus0/target0/lun0/part3
mkfs.ext2 /dev/loop/0
aeslosetup -d /dev/loop/0
aeslosetup -e AES256 /dev/loop/1 /dev/scsi/host0/bus0/target0/lun0/part4
mkfs.ext2 /dev/loop/1
aeslosetup -d /dev/loop/1

#aeslosetup -e AES256 /dev/loop/0 /dev/scsi/host0/bus0/target0/lun0/part3
#mount -t ext2 /dev/loop/0 /mnt/part2-crypted/
#aeslosetup -e AES256 /dev/loop/1 /dev/scsi/host0/bus0/target0/lun0/part4
#mount -t ext2 /dev/loop/1 /mnt/part3-crypted/
```

Known Error:

Error:

```
/dev/loop/0: No such file or directory
```

In /etc/init.d/boot there is a line similar to this:

```
load_modules /etc/modules.d/*
```

That is called before the /mnt/part1 is mounted.

Solution 1:

```
insmod /mnt/part1/jffs_export/lib/modules/2.4.35.4/loop.o
```

Solution 2:

Put this command into a later called shell-script.

E.G.:

```
echo 'mount /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/part1' >> /etc/init.d/externalmount
echo 'insmod /mnt/part1/jffs_export/lib/modules/2.4.35.4/loop.o' >> /etc/init.d/externalmount
chmod +x /etc/init.d/externalmount
# ToDo JHR
```

```
echo "
config mount
    option target    /mnt/part2-crypted
    option device    /dev/scsi/host0/bus0/target0/lun0/part3
    option fstype    ext2
    option options   defaults,loop=/dev/loop/0,encryption=AES256
    option enabled   0
```

```
config mount
    option target    /mnt/part3-crypted
    option device    /dev/scsi/host0/bus0/target0/lun0/part4
    option fstype    ext2
    option options   defaults,loop=/dev/loop/1,encryption=AES256
    option enabled   0" >> /etc/config/fstab
```

```
mount /mnt/part2-crypted
#(Passworteingabe)
mount /mnt/part3-crypted
#(Passworteingabe)
```

```
umount /mnt/part2-crypted;
umount /mnt/part3-crypted;
aeslosetup -d /dev/loop/0
aeslosetup -d /dev/loop/1
```

5.3.7 Samba

Zunächst muss die Paket-Liste aktualisiert werden um danach das Samba-Paket installieren zu können:

```
opkg update
opkg -dest usb install samba-server
opkg-link.sh add samba-server
opkg list_installed
```

```
mkdir /mnt/part1/etc
mkdir /mnt/part1/log
```

In der Datei `"/etc/samba/smb.conf"` kann der Ordner der freigegeben werden soll definiert werden. Standardmäßig ist das der `"/tmp"`-Ordner. Alle Änderungen (hochgeladene

Dateien) gehen also bei einem Reset verloren. Mit `"/etc/init.d/samba start"` kann der Samba-Server gestartet werden. Meine `"/etc/samba/smb.conf"` kann man sich folgendermaßen herunterladen:

```
cd /etc/samba
rm smb.conf
wget http://www.helber.it/fileadmin/Projekte/Router/smb.conf
```

```
echo "[global]
syslog = 0
syslog only = yes
workgroup = OpenWrt
server string = OpenWrt Samba Server
security = share
browseable = yes
guest only = no
log level = 1
log file = /mnt/part1/log/samba.log
max log size = 100
smb passwd file=/mnt/part1/etc/smbpasswd
encrypt passwords = yes
guest account = nobody
local master = yes
name resolve order = lmhosts hosts bcast
dns proxy = no
```

```
[public]
comment = part3-crypted/public
path = /mnt/part3-crypted/public/
browseable = yes
public = yes
writeable = no
```

```
[john]
comment = part3-crypted/john
path = /mnt/part3-crypted/john
browseable = yes
public = yes
writeable = no
valid users = john
```

```
[upload]
comment = part3-crypted/upload
path = /mnt/part3-crypted/upload/
browsabke = yes
public = yes
```

```
writeable = yes
#valid users = root" > /mnt/part1/jffs_export/etc/samba/smb.conf

ln -s /mnt/part1/jffs_export/etc/samba/smb.conf /etc/samba/smb.conf

# This will add a user john with password john as system user
echo "john:$1$cenDCinJ$cm2lKiOU0A7A9A4wkIs3B0:1000:1000:nobody:/var:/bin/false" >> /e

# Change the owner of the folder
chown john /mnt/part3-crypted/john

# Change the password
passwd john

# Now we add the user john as a samba user as well
smbpasswd --help
smbpasswd -a john
```

Replace:

```
127.0.0.1 localhost.
```

By:

```
127.0.0.1 localhost. OpenWrt
```

5.3.8 Performance

Samba über WLAN 1385 kB/s

5.3.9 Cron

Precondition: Section [5.3.13](#) on page [27](#).

<http://www.macsat.com/macsat/content/view/20/30/>

The cron daemon (crond) is a standard part of OpenWRTs version of BusyBox. This means that there is nothing to install, just a few things to setup.

Since crontab can be a pain to use, lets create a few dirs, where we can place scripts that can be ran at specified times:

```
mkdir /etc/cron.5mins
mkdir /etc/cron.hourly
mkdir /etc/cron.daily
mkdir /etc/cron.weekly
mkdir /etc/cron.monthly
```

Now you need to create the file: `/etc/crontabs/root` like this:

```
echo "# Syntax for lines is : minute hour day month dayofweek command #
*/5 * * * * /usr/bin/run-parts /etc/cron.5mins
15 * * * * /usr/bin/transfer.sh
01 * * * * /usr/bin/run-parts /etc/cron.hourly
02 4 * * * /usr/bin/run-parts /etc/cron.daily
22 4 * * 0 /usr/bin/run-parts /etc/cron.weekly
42 4 1 * * /usr/bin/run-parts /etc/cron.monthly" > /etc/crontabs/root
```

As also noted in the file, the syntax for each line is :
minute hour day month dayofweek command

To take an example, the command: `/usr/bin/run-parts /etc/cron.weekly` is executed at time :
minute : 22
hour : 4
day : *
month : *
dayofweek : *

Where "*" is a wild-card meaning "ALL".

Thus the command is running : Every day of week, Every Month, Every Day at 4:22
Note that cron uses a 24h clock.

Having created this config-file, all that is needed is the command "run-parts" which is not a part of the firmware.

For this purpos I created a small script which I named run-parts and placed in `/usr/bin`
A script like this will do the job:

```
cd /mnt/part1/bin/
wget http://www.helber.it/fileadmin/Projekte/Router/run-parts
# Remember to chmod the script to be executable :
chmod +x /mnt/part1/bin/run-parts
ln -s /mnt/part1/bin/run-parts /usr/bin/run-parts
ls -la /usr/bin/run-parts
```

All that is left to do, is to ensure the cron system get stated at boot-time. Create the file `/etc/init.d/S61crond`:

```
echo "#!/bin/sh /etc/rc.common
# Copyright (C) 2006 OpenWrt.org
START=50

start () {
    [ -z $(ls /etc/crontabs/) ] && exit 1
```

```

    mkdir -p /var/spool/cron
    [ -L /var/spool/cron/crontabs ] || ln -s /etc/crontabs /var/spool/cron/crontabs
    crond -c /etc/crontabs
}

stop() {
    killall -9 crond
}" > /etc/init.d/cron
# Make the file executable:
chmod +x /etc/init.d/cron
ln -s /etc/init.d/cron /etc/rc.d/S50cron

```

In order to test the cron-system, you could try to make a small script in `/etc/cron.5mins` called `test.sh`, like this:

```

echo "#!/bin/sh
date >> /tmp/crontest.txt" > /etc/cron.5mins/test.sh
# Remember to make it executable:
chmod 755 /etc/cron.5mins/test.sh

```

Try to reboot your router (or call `"crond -c /etc/crontabs -b"`) and let it run for some 10 or 15 minutes. Then you should see a list of timestamps in `/tmp/crontest.txt` showing when the cron has been running the script:

```
cat /tmp/crontest.txt
```

To let the cron-daemon call the synchronisation daily:

```
ln -s /usr/bin/transfer.sh /etc/cron.daily/transfer.sh
```

5.3.10 DynDns

Precondition: Section [5.3.9](#) on page [23](#).

```

cd /etc/cron.5mins/
wget http://www.helber.it/fileadmin/Projekte/Router/dyndns.sh.txt
mv dyndns.sh.txt dyndns.sh
chmod +x dyndns.sh

```

5.3.11 Rsync

```

opkg update
opkg -dest usb install rsync
opkg-link.sh add rsync

```

```
opkg-link.sh add libpopt
# Without a swap-partition only very few files can be transfered
opkg install swaputils

opkg install -dest usb dropbearconvert
opkg-link.sh add dropbearconvert
dropbearconvert openssh dropbear /mnt/part3-rypted/siemens-key \
  /mnt/part3-rypted/siemens-key-dropbear

mkdir /mnt/part1/tmp
```

That's how my backup-script looks like:

```
cd /mnt/part1/bin/
wget http://www.helber.it/fileadmin/Projekte/Router/transfer.sh
chmod +x transfer.sh
ln -s /mnt/part1/bin/transfer.sh /usr/bin/transfer.sh
ls -la /usr/bin/transfer.sh
```

Since rsync first exchanges the filelist from one system to the other, usually ram and swap are used heavily if partitions with lots of files are being synced:

```
echo "while true
do
clear
free
sleep 5
done" > /mnt/part1/bin/freeloop.sh
chmod +x /mnt/part1/bin/freeloop.sh
ln -s /mnt/part1/bin/freeloop.sh /usr/bin/freeloop.sh

# Can only be terminated from a different console with ‘killall sh’
freeloop.sh
```

5.3.12 Zeitzone

Mit folgendem Befehl stellen wir die Zeitzone auf Berlin/Germany um:

```
echo "CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00" > /etc/TZ
# Howto enter the correct date and time?
date --help
# e.g.: date '2009-03-04 23:15'
```

5.3.13 RDate

```
echo "#!/bin/sh
/usr/sbin/rdate -s sunsite.auc.dk
echo \"CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00\" > /etc/TZ" > /etc/init.d/rdate
# Remember to make it executable
ln -s /etc/init.d/rdate /etc/rc.d/S60rdate
chmod +x /etc/init.d/S60rdate
```

Now running `/etc/S60rdate` or rebooting your router, will set the time correctly according to your current timezone.

Now we are ready to configure cron.

5.3.14 OpenVPN

Precondition: Section [5.3.13](#) on page 27.

[[OpenWrt Wiki OpenVPN](#)] [[DDWrt Wiki OpenVPN](#)] First of all OpenVPN needs to be installed:

```
opkg -dest usb install openvpn
opkg-link.sh add openvpn
opkg-link.sh add kmod-tun
opkg-link.sh add libopenssl
opkg-link.sh add zlib
opkg-link.sh add liblz

mkdir /mnt/part1/OpenVPN
cd /mnt/part1/OpenVPN
openvpn --genkey --secret static.key
ln -s /mnt/part1/OpenVPN/static.key /etc/static.key

echo "
config rule
    option src            wan
    option dest_port     1194
    option target        ACCEPT
    option proto         udp

config rule
    option src            wan
    option dest_port     1194
    option target        ACCEPT
    option proto         tcp" >> /etc/config/firewall
uci commit firewall
/etc/init.d/firewall restart
```

```
mkdir /etc/openvpn
cd /etc/openvpn
wget http://www.helber.it/fileadmin/Projekte/Router/openvpn-startupscript
mv openvpn-startupscript startupscript
cd /mnt/part1/OpenVPN
wget http://www.helber.it/fileadmin/Projekte/Router/server.ovpn

ln -s /mnt/part1/OpenVPN/server.ovpn /etc/openvpn/server.ovpn
ln -s /mnt/part1/OpenVPN/static.key /etc/openvpn/secret.key
iptables -A INPUT -i tap+ -j ACCEPT
iptables -A FORWARD -i tap+ -j ACCEPT
chmod +x /mnt/part1/OpenVPN/startupscript

echo "config interface vpn
      option ifname tap0
      option proto static
      option netmask 255.255.255.0
      option dns ''
      option gateway ''
      option ipaddr 10.10.10.1" >> /etc/config/network

echo "
config 'zone'
      option 'name' 'vpn'
      option 'input' 'ACCEPT'
      option 'output' 'ACCEPT'
      option 'forward' 'ACCEPT'

config 'forwarding'
      option 'src' 'lan'
      option 'dest' 'vpn'" >> /etc/config/firewall

echo "#!/bin/sh /etc/rc.common

START=65
STOP=35

start() {
    /etc/openvpn/startupscript up
    openvpn --daemon --config /etc/openvpn/server.ovpn
}

restart() {
    $0 stop
    sleep 3
    $0 start
```

```

}

reload() {
    killall -SIGHUP openvpn
}

stop() {
    killall openvpn
    sleep 3
    /etc/openvpn/startupscript down
}" > /etc/init.d/openvpn2
chmod +x /etc/init.d/openvpn2

vi /etc/config/firewall
/etc/init.d/firewall restart
uci show network; uci show firewall
/mnt/part1/OpenVPN/startupscript up
ifconfig tap0 10.10.10.1 netmask 255.255.255.0
route add -net 192.168.4.0 netmask 255.255.255.0 gw 10.10.10.2 tap0
# LAN Steffen: 192.168.4.0/24
openvpn --daemon --config /etc/openvpn/server.ovpn

#brctl show
#brctl addif br-lan tap0
#ifconfig tap0 0.0.0.0 promisc up

```

Known problems:

First decrypt, then encrypted transfer, than encrypt

Better transfer stream of filesystem ... but different clusters might be corrupt on the different hdd

5.3.15 Socks5

```

opkg -dest usb install srelay opkg-link.sh add srelay echo "# allow local subnet to access
socks proxy 192.168.1.0/24 any - 127.0.0.1/32 any -" > /etc/srelay.conf srelay -h srelay
-c /etc/srelay.conf -r -s

```

5.3.16 WOL

```

opkg -dest usb install etherwake opkg-link.sh add etherwake

```

```

echo "etherwake 00:30:05:8a:51:ff" > /bin/wol.sh chmod +x /bin/wol.sh wol.sh ping
192.168.1.240

```

5.3.17 QoS

Um QoS⁶ nutzen zu können, müssen wir TC installieren und ein paar Daten auf den Router kopieren:

```
ipkg update

# Da die Installation von TC scheinbar extrem viel
# System-Ressourcen benötigt, schaufeln wir ein paar frei
killall httpd
killall crond
killall udhcpc

ipkg install kmod-sched kmod-ipt-contrack iptables-mod-contrack kmod-ipt-ipt \
iptables-mod-ipt kmod-ipt-extra iptables-mod-extra iptables-extra tc iptables-mod-filter

#ipkg install http://www.helber.it/fileadmin/Projekte/Router/qos-scripts_0.9.1-1_mipsel.ipk
#ipkg install http://www.helber.it/fileadmin/Projekte/Router/tc_2.6.11-050330-1_mipsel.ipk

cd /etc/l7-protocols/
wget http://www.helber.it/fileadmin/Projekte/Router/pattern.tar
tar -xvf pattern.tar
rm pattern.tar

cd /usr/bin
# Danach richten wir uns einige nette Shell-Skripte ein,
# welche uns einige coole Features einrichten:

# http://www.ipcop-forum.de/forum/viewtopic.php?t=4749&postdays=0&postorder=asc&start=120
# http://voip-info.org/tiki-index.php?page=QoS%20Linux%20with%20HFSC
# http://www.ip-phone-forum.de/archive/index.php/t-87301.html
wget http://www.helber.it/fileadmin/Projekte/Router/qos.sh

#wget http://www.helber.it/fileadmin/Projekte/Router/qos2.sh

# SipShaper: QoS mit HFSC - bei mir klappt's super! (HFSC)
# http://www.ip-phone-forum.de/showthread.php?t=72590
wget http://www.helber.it/fileadmin/Projekte/Router/qos3.sh

wget http://www.helber.it/fileadmin/Projekte/Router/qos4.sh
wget http://www.helber.it/fileadmin/Projekte/Router/qos5.sh

# HTB
# http://www.ip-phone-forum.de/showthread.php?t=95566
wget http://www.helber.it/fileadmin/Projekte/Router/load_shaper_modules
wget http://www.helber.it/fileadmin/Projekte/Router/qos6.sh

# NSLU2
# http://www.nslu2-linux.org/wiki/HowTo/EnableTrafficShaping
# Verweis von: http://www.ip-phone-forum.de/showthread.php?p=344900#post344900
wget http://www.helber.it/fileadmin/Projekte/Router/qos7.sh

chmod +x /usr/bin/*.sh

# ordentliche tc-Version die auch HTB kann
# im Repository war beim schreiben der Doku V2.0 aktuell
#wget http://www.helber.it/fileadmin/Projekte/Router/htb3.6-020525.tgz
#gzip -d htb3.6-020525.tgz
#tar -xvf htb3.6-020525.tar
```

Um den Wert MYUMAX in der Datei "/usr/bin/qos.sh" eintragen zu können, muss dieser erst ermittelt werden. Dazu gibt man auf der Konsole folgende Kommandos ein:

⁶Quality of Service

```

i=1400
o=0
while [ $o -eq 0 ]
do
  echo $i
  ping -s $i -c 1 ftp.t-online.de > /tmp/ping
  o='cat /tmp/ping | grep -c '0 packets received''
  i='expr $i + 1'
done
i='expr $i + 26'
echo Es muss $i eingetragen werden!
rm /tmp/ping

```

Der Wert UPRATEJAN in der Datei `"/usr/bin/qos.sh"` muss auf den maximalen Upload (kBit) eingestellt werden.

Um zu überprüfen welche Pakete wie markiert werden kann man folgenden Befehl verwenden:

```

#wget http://www.tom-e.de/download/iftop016.tar.gz
#tar xzf iftop016.tar.gz
#cd iftop016
#./install -i
#iftop -i ppp0

```

```
grep 192.168.1.100 /proc/net/ip_contrack
```

5.4 Shell-Skripte

Wenn man folgende Befehle der Reihe nach kopiert und in der Konsole einfügt, erhält man diese Funktionalität dazu:

- `whatismyip.sh`: Gibt die IP-Adresse des Routers aus
- `switchmode.sh {switch | router | routerwan | lanparty}`
Ich nutze im Wohnheim den Modus `routerwan`. Einstellungen am Modus kann man in der Datei `"/etc/init.d/S02mode_switch"` vornehmen. Man sollte natürlich wissen was man tut ;)
- deutlich einfacheres Verwalten der Firewallregeln!
Es ist zwar möglich Änderungen an der Datei `"/usr/bin/firewall.user.sh"` vorzunehmen. Allerdings wird diese Datei bei jedem Start des Routers durch das `"/etc/init.d/S49startup"-` Skript aufgerufen. Sobald mal ein Fehler in der `"firewall.user.sh"` gemacht wurde bringt auch ein Reset des Routers nichts mehr. Daher empfehle ich Änderungen zunächst in der Datei `"firewall.user.test.sh"` zu machen und diese dann manuell auszuführen (TESTEN!). Den Grundgedanken dazu habe ich von [[Simple Firewall](#)]. Ich habe allerdings verschiedenes angepasst und verbessert.
Die Firewall-Konfiguration lässt sich mit folgenden Befehlen anzeigen:

```

iptables -t nat -L
iptables -L

```

- Cron-Jobs werden möglich gemacht. Ein Cron-Job wird eingerichtet, der stündlich die Uhrzeit übers Netz aktualisiert.
- Aliase werden angelegt, die das Arbeiten mit dem System vereinfachen sollen.
- Feste Hosteinträge für den DNS werden gesetzt.
- cooles Skript um Dateien auf Linux-Rechner zu übertragen "tarsend.sh"
- cooles Skript um Dateien von Linux-Rechner zu übertragen "tarrecv.sh"
- cooles Skript um die Signalstärke zu einem verbundenen Client zu testen "signalstaerke.sh"

Diese Befehle einfach kopiert und in der Konsole eingefügt.

```

cd /usr/bin
wget http://www.helber.it/fileadmin/Projekte/Router/signalstaerke.sh
wget http://www.helber.it/fileadmin/Projekte/Router/tarsend.sh
wget http://www.helber.it/fileadmin/Projekte/Router/tarrecv.sh
wget http://www.helber.it/fileadmin/Projekte/Router/whatismyip.sh
wget http://www.helber.it/fileadmin/Projekte/Router/switchmode.sh
wget http://www.helber.it/fileadmin/Projekte/Router/fwlib.sh
wget http://www.helber.it/fileadmin/Projekte/Router/firewall.user.sh
wget http://www.helber.it/fileadmin/Projekte/Router/firewall.user.test.sh
cd /etc/init.d
wget http://www.helber.it/fileadmin/Projekte/Router/S49startup
wget http://www.helber.it/fileadmin/Projekte/Router/S02mode_switch
cd /etc
wget http://www.helber.it/fileadmin/Projekte/Router/functions.sh
rm banner
#Und wenn man noch eine schöne Begrüßung nach dem Einloggen haben möchte:
wget http://www.helber.it/fileadmin/Projekte/Router/banner

touch /usr/sbin/cron.minutely
touch /usr/sbin/cron.hourly
touch /usr/sbin/cron.daily
chmod +x /usr/sbin/cron.*
rm /etc/init.d/S60cron
echo "#!/bin/sh" > /etc/init.d/S60cron
echo "/usr/sbin/crond" >> /etc/init.d/S60cron
chmod +x /etc/init.d/S60cron

#echo "#!/bin/sh" > /usr/bin/whatismygw.sh
#echo "netstat -rn | awk '/^0\.0\.0\.0/ {print \$2}'" >> /usr/bin/whatismygw.sh
#chmod +x /usr/bin/whatismygw.sh
#TODO netstat

#echo "#!/bin/sh" > /usr/bin/whatismygwmac.sh
#echo "arping -fq -I vlan1 `whatismygw.sh`" >> /usr/bin/whatismygwmac.sh
#echo "cat /proc/net/arp | grep `whatismygw.sh` | awk '{print \$4}'" \
# >> /usr/bin/whatismygwmac.sh
#chmod +x /usr/bin/whatismygwmac.sh

echo "[ -f /usr/bin/firewall.user.sh ] && . /usr/bin/firewall.user.sh" >> /etc/init.d/S45firewall
echo "iptables -I INPUT 8 -p tcp --dport 22 -j ACCEPT" >> /etc/init.d/S45firewall

#neu angelegte Skripte ausführbar machen
chmod +x /usr/bin/*.sh
chmod +x /etc/init.d/S49startup
chmod +x /etc/init.d/S02mode_switch
./S49startup

```

Hier gibts eine kleine Zusammenfassung über die verschiedenen Modi:

- Switch mode:
 - WAN deaktiviert, Alle 5 Ports werden fürs LAN genutzt
 - LAN wird per DHCP konfiguriert
 - Das WLAN-Interface wird zum LAN gebridged
WPA2 Verschlüsselung aktiviert
 - Firewall deaktiviert
 - DHCP server deaktiviert
 - OpenWRT web interface deaktiviert
- Router-Modus:
 - WAN aktiviert (PPPoE)
 - LAN aktiviert: nutzt statische IPs (beim Client einstellen)
 - WLAN-Konfiguration wie im Switch-Modus
 - Alle Dienste aktiviert
- RouterWAN-Modus:
 - WAN aktiviert (DHCP)
 - LAN-Konfiguration wie beim Router-Modus
 - WLAN deaktiviert
 - Firewall aktiviert
 - DHCP server aktiviert
 - OpenWRT web interface aktiviert
- Lanparty-Modus:
 - WAN disabled
 - LAN-Konfiguration wie im Router-Modus
 - WLAN aktiviert: anders konfiguriert als in den vorherigen Modi
 - Firewall deaktiviert
 - DHCP server aktiviert
 - OpenWRT web interface deaktiviert
 - spezieller LANparty web server (httpd2): z.B. zum Pizza bestellen ;)

5.5 MAC address cloning

Bei mir im Wohnheim ist für meinen Anschluss nun dummerweise die Mac-Adresse meines Notebooks eingetragen. Ausschließlich die Mac-Adresse "00:03:0D:02:1D:7A" hat Zugriff auf das Internet. Da es aber schon etwas spät war den Admin raus zu klingeln und ich unabhängig davon auch gar keine Lust hatte ihm zu erzählen, dass ich ab sofort einen Router am Netz habe, hab ich mich über "MAC address cloning in OpenWrt" schlau gemacht. Das funktioniert folgendermaßen:

Um die Mac-Adresse des Routers zu ändern müssen die Variablen "wan_hwaddr" und "et0macaddr" im NVRAM überschrieben werden. Das funktioniert mit folgenden Kommandos:

```
#OpenWrt Kamikaze
ifconfig eth0 ether 00:11:22:33:44:55
# ToDo JHR

# OpenWrt White russian
nvram set wan_hwaddr="00:03:0D:02:1D:7A"
nvram set et0macaddr="00:03:0D:02:1D:7A"
nvram show | grep mac
nvram commit

ifdown wan
ifup wan
```

Da mein LapTop (Windows XP) natürlich nicht die selbe MAC-Adresse haben kann wie der Router (sonst funktioniert gar nichts mehr) musste ich diese auch noch ändern unter "Start > Systemsteuerung > System > Hardware > Geräte-Manager > Netzwerkadapter > [Kartename] > Eigenschaften > Erweitert > Network Address". Dort stellte ich einfach "00030D021D7B" ein.

Oder unter Linux:

```
ifconfig eth0 ether 00:03:0D:02:1D:7B
```

Und wenn wir schon dabei sind, tragen wir im Router für die entsprechende MAC-Adresse eine feste IP ein:

5.6 SSH-Public-Key-Authentication

Um nicht jedes mal das Passwort eingeben zu müssen, kann man sich wie bei jedem Linux mit einem Key authentifizieren⁷. Zum erstellen solcher Keys ist unter Windows das Tool "puttygen.exe" gut geeignet.

⁷wobei man den Public Key ohne Risiko veröffentlichen kann

Der generierte Public-Key muss dann unter `"/etc/dropbear/authorized_keys"` und `"/etc/dropbear/id_rsa.pub"` abgelegt werden. Z.B. so:

```
echo "ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQAgBxCRPq4/nc08NFEzWF8y17/jWBorP"\
"WKb/nyZs+bcq/pa5x23tZ000Mb73tzN5APq1h1xr7cUzQLajSMHm3p0Ie6s1bamQqwgZ"\
"TL012oSvqgndqjmJM3A7hdi4yHb6A0hiVEa20I1LWCsJPnqpZmOCsTjurtJH2QNN7Wp3"\
"v2ycaD+gdazIf5wtq8dBRE4EE8pAba88yhr3/Gi/Wqc3N1UGRSMDp04xosm1SFT1Dc4d"\
"wV3zWf1PFop1SErf+8L18xCfTS6hpz42bwRcbRd3wbEgvRoV4v0ZVCjL+R9n1L79Kv9b"\
"2WuR59IK72ke58D1Q9m1HxVr53xSoK+ToFSazoVLwt9AeCQbtKJNUZrI1c70b6GAERm1"\
"ANGr4qqsdDEcu219u12sbpkXzrP1j+GXVLq6mzR3vIP44MwEoSABPik/NCrekjbdNgvE"\
"CsvZ86KyR6Hf8KvyALyTSBuo/sAJIPe0H1AZ/8DDr5/uDt07EJL82prIOLnLyMKUqhs"\
"EZ68LUFouM/c4RMsT9QHmYtNMYRtked1MOKHUXsEWgm65Re1ozMDNIKm7jgwMyXTbrXC"\
"CZCeZnd8ca07fpCyT5z6ivFh9jqvdRa7rDK2QTPISXRDFZ11jvdYkmIbnd6BaB4OrWmf"\
"NuNO/WY7iMcNAJqRK91M12UyYohcRd11jaWur"\
"5YGrwQ== rsa-key-20081228" >> /etc/dropbear/authorized_keys
chmod 0600 /etc/dropbear/authorized_keys
```

6 Netzwerk Konfiguration

With the password (user: root) set in section 5.2 on page 8 it is also possible to log in on the "OpenWRT Admin Console" (WebInterface) usually accessible on <http://192.168.1.1>.

- LAN: vlan0
- WAN: vlan1
- WiFi: eth1

<http://wiki.openwrt.org/OpenWrtNVRAM>

6.1 WiFi

```
config wifi-device wlo
    option type      broadcom
    option channel 1

    # REMOVE THIS LINE TO ENABLE WIFI:
    option disabled 0

config wifi-iface
    option device    wlo
    option network   lan
    option mode      ap
    option ssid      wellenreiter4
```

```
option encryption psk
option key "gbGKqGn7QaWxjgXeuhkb"
```

Das Ändern von Variablen im RAM erfolgt immer auf diese Art und Weise.

```
nvrnram get variable
nvrnram set variable=value
```

```
# Dieser Befehl schreibt die Einstellungen vom RAM im Flash fest,
# damit die Einstellungen durch einen Reboot nicht verloren gehen.
nvrnram commit
# maximal 1.000-10.000x schreiben auf Flash möglich (sparsam verwenden)
```

In nachfolgender Tabelle stehen die grundlegenden Variablen um das Wireless-LAN zu konfigurieren:

NVRAM	VARIABLE DESCRIPTION
wl0_radio	<i>Enable / disable the radio (1=enable)</i>
wl0_mode	<i>ap = Access Point (master mode) sta = Routing client mode wet = Bridged client mode</i>
wl0_ssid	<i>ESSID</i>
wl0_infra	<i>0 = Ad Hoc mode 1 = normal AP/Client mode</i>
wl0_closed	<i>0 = Broadcast ESSID 1 Hide ESSID</i>
wl0_channel	<i>1 / 2 / 3 /.../ 11 channel</i>
wl0_macmode	<i>(disabled/allow/deny) used to (allow/deny) mac addresses listed in wl0_maclist</i>
wl0_maclist	<i>List of space separated mac addresses to allow/deny according to wl0_macmode. Addresses should be entered with colons, e.g.: "00:02:2D:08:E2:1D 00:03:3E:05:E1:1B". if you have more than one mac use quotes, or only the first will be recognized.</i>

Table 1: WiFi NVRAM

Möchte man die Übertragung verschlüsseln (WEP WPA) wird man auf folgender URL bei Kapitel 4.3 fündig: <http://wiki.openwrt.org/OpenWrtDocs/Configuration>
Mit folgenden Befehlen erlaubt man z.B. nur den angegebenen MAC-Adressen das WLAN zu nutzen:

```
nvrnram set wl0_maclist='00:03:0D:02:1D:7A 00:03:0D:02:1D:7B'
nvrnram set wl0_macmode=allow
nvrnram commit
```

6.2 LAN

Wegen der Art und Weise wie die Schnittstellen in der Hardware realisiert sind (eine Schnittstelle, mehrere LAN-Ports), sind für das Gerät erweiterte vlan Einstellungen erforderliche. Wenn diese nicht auf die korrekten Werte eingestellt sind, werden die Schnittstellen nicht richtig zugewiesen. Info: Wenn Sie "admcfg" oder ähnliches benutzen, kann es sein, dass dieses Problem Sie nicht betrifft (ich bin nicht sicher).

Versichern Sie sich, dass die NVRAM Einstellungen den empfohlenen Werten entsprechen:

NVRAM	EMPFOHLENER WERT DER EINSTELLUNG
vlan0hwname	<i>et0</i>
vlan0ports	<i>1 2 3 4 5*</i>
vlan1hwname	<i>et0</i>
vlan1ports	<i>0 5</i>

Table 2: LAN NVRAM

Das heißt, eine Schnittstelle mit der Bezeichnung "vlan0", ist mit den LAN-Ports 3-0 des internen Switches verbunden (normalerweise mit "LAN 1-4" auf dem Kasten beschriftet), und eine Schnittstelle mit der Bezeichnung "vlan1" ist mit dem LAN-Port 4 des internen Switches verbunden (normalerweise mit "WAN" auf dem Kasten beschriftet). LAN-Port 5 des internen Switches trägt alle VLANs markiert (wird durch das Sternchen bewirkt) zur realen Schnittstelle et(h)0.

PHYSICALLY:

```

          tagged      +-----+
eth0 ===== | 5      SWITCH      |
               | 4    3    2    1    0 |
               +-----+
               |   |   |   |   |   |
               WAN  LAN1 LAN2 LAN3 LAN4

```

LOGICALLY:

```

vlan0 ----- LAN 1-4
vlan1 ----- WAN

```

Wenn das NVRAM mit diesen Werten eingestellt ist, dann sind die empfohlenen Werte für "wan_ifnames" und "lan_ifnames" korrekt. Info: Es ist möglich Änderungen an den LAN-Ports vorzunehmen (auch welcher LAN-Port der WAN-Port ist usw), aber das ist im allgemeinen keine gute Idee.

7 Backup

After all this work it is time to save a backup of the system. Therefore execute the following command on a linux workstation which is connected with the router via ethernet:

```
IP=192.168.1.1
ssh root@$IP -C 'mount -o remount,ro /dev/mtdblock/4 \
/jffs ; dd if=/dev/mtdblock/1 ; mount -o remount,rw \
/dev/mtdblock/4 /jffs' > 2009-10-21-router-backup.trx
```

After a system-crash the router can be restored easily:

```
IP=192.168.1.1
sudo atftp --tftp-timeout 1 --trace -p -l image.trx $IP
```

8 Information about Siemens SE 505

- Platform & frequency: Broadcom 4712 @ 200MHz
- Flash: 4MB
- RAM: 8MB
- Switch: ADM6996L
- Wireless NIC: Broadcom (integrated)

9 Viel Spaß!

Ich wünsche viel Spaß mit dem Router!

Wenn ein Leser dieser Dokumentation erfolgreich eine WebCam, ein Bluetooth-Gerät oder etwas ähnliches an dem Router anschließt, wäre ich über eine kurze Info (wenn möglich kurze Anleitung) dankbar. Ich würde diese Infos dann in dieser Anleitung aufnehmen. In meinem Wohnheim sollte der Router über den WAN-Port auch Dateifreigaben weiter leiten. Wenn ich mal Zeit und Lust habe, werde ich das Problem lösen und hier dokumentieren.

10 FAQ

Was tun wenn man richtig Blödsinn gebaut hat und nichts mehr funktioniert?!
Dann entfernt man die Spannungsversorgung des Routers, wartet kurz und steckt sie wieder ein, während man die RESET-Taste gedrückt hält. Daraufhin flimmert die WLAN-LED am Router kurze Zeit sehr schnell. Dabei werden alle Einstellungen auf den Ausgangszustand zurück gesetzt (auch boot_wait *g*). Sobald der Router neu startet (dass

erkennt man daran, dass Die LEDs von WAN & LAN1-4 kurz aufleuchten), spielt man per TFTP ein frisches Image ein. Danach sollte wieder ein Zugriff per SSH oder Telnet möglich sein.

Was tun wenn jedesmal nach dem Start von firstboot das System am A**** ist?
Kopier dir die Datei `"/rom/bin/firstboot"` nach `"/"`⁸ und lösche die 64. Zeile. `"test -f etc/dropbear/dropbear_dss_host_key && rm etc/init.d/S*telnet"`. Dadurch bleibt der passwortfreie Telnetzugang zum System immer erhalten. Wenn das System wieder stabil bootet und man sich per SSH einloggen kann löscht man die Datei `"/etc/init.d/S*telnet"` einfach von Hand⁹.

11 Anhang

12 Quellen

References

- [HowTo OpenWRT] <http://wiki.openwrt.org/CategoryHowTo?action=show&redirect=OpenWrtHowTo>
- [HowTo OpenNet] http://wiki.opennet-initiative.de/index.php/Siemens_SE505
- [Simple Firewall] <http://wiki.openwrt.org/SimpleFirewall>
- [Port Weiterleitung] <http://wiki.opennet-initiative.de/index.php/Portforwarding>
- [USB auflöten] <http://wiki.openwrt.org/OpenWrtDocs/Customizing/Hardware/USB>
- [USB storage howto] <http://wiki.openwrt.org/UsbStorageHowto>
- [WLAN Clientmode] <http://wiki.openwrt.org/ClientModeHowto>
- [IP publishing] <http://wiki.openwrt.org/PublishYourWANIp>
- [OpenWrt Hardware] <http://wiki.openwrt.org/OpenWrtDocs/Hardware/Siemens/SE505>
- [Filesystem Encryption on OpenWrt] <http://loblog.wordpress.com/2008/05/26/filesystem-encryption-on-openwrt-kamikaze-kernel-24/>

⁸`cp /rom/bin/firstboot /`

⁹`rm /etc/init.d/S*telnet`

[OpenWrt Wiki OpenVPN] <http://wiki.openwrt.org/OpenVPNHowTo>

[DDWrt Wiki OpenVPN] [http://www.dd-wrt.com/wiki/index.php/
OpenVPN-Tunnel_Server/_Client,_Netzwerke_verbinden#
Server_Config_.28WRT54_-_LINUX.29](http://www.dd-wrt.com/wiki/index.php/OpenVPN-Tunnel_Server/_Client,_Netzwerke_verbinden#Server_Config_.28WRT54_-_LINUX.29)

13 Marken

Diese Dokumentation beinhaltet eingetragene Marken oder Marken der jeweiligen Eigentümer.

Einige Firmen- und/oder Produktbezeichnungen in dieser Dokumentation sind Warenzeichen und/oder eingetragene Warenzeichen ihrer jeweiligen Besitzer.

14 Haftungsausschluss

Die Vollständigkeit und verlässlichkeit der in dieser Dokumentation enthaltenen Informationen wurden sorgfältig überprüft. Für die Richtigkeit der Angaben kann ich jedoch keine Gewähr übernehmen und für eventuelle Schäden nicht haften. In keinem Fall bin ich Ihnen gegenüber haftbar für Schäden, die auf die Verwendung oder den anderweitigen Einsatz dieser oder anderer Dokumentationen zurückzuführen sind.

Figure 6: Gehäuse (angelötete USB-Buchse)



Figure 7: Gehäuse (von unten)

