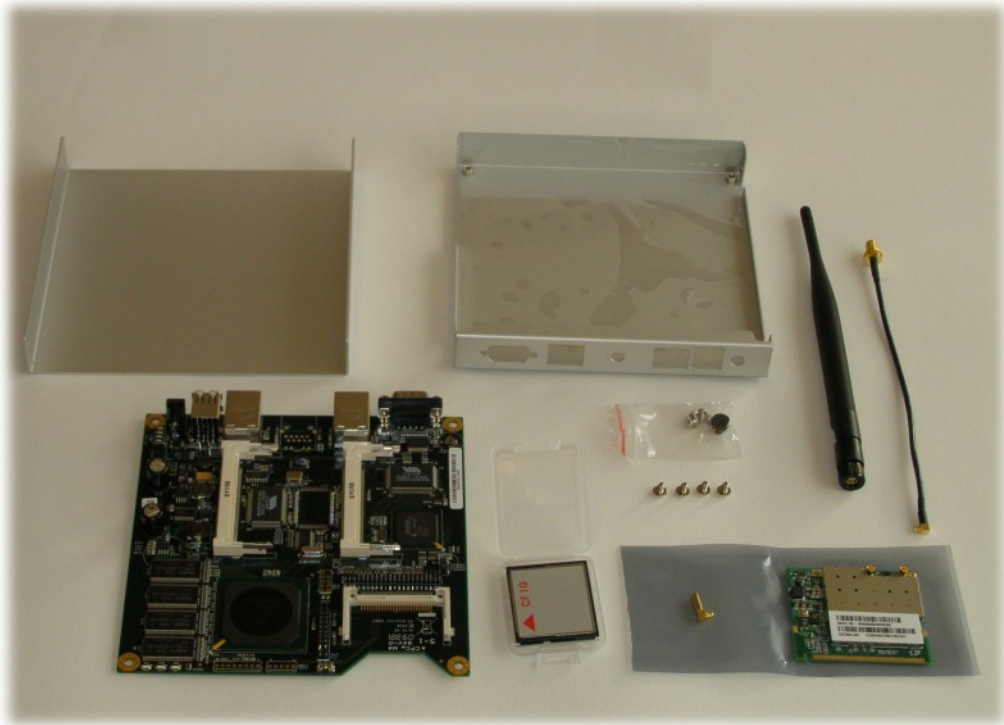


HAMNET ALIX - HowTo get started



Thomas Ries

HAMNET ALIX - HowTo get started

von Thomas Ries

Copyright © 2010-2011 Thomas Ries

Dieses Dokument steht unter der GNU Public Lizenz (GPL) und darf gemäss derer Bestimmungen frei verteilt werden.

Personen, die zu diesem Dokument beitragen stimmen zu, dass ihr Beitrag unter der GPL veröffentlicht wird.

Versionsgeschichte

Version 1 2010-02-21 Geändert durch: hb9xar@uska.ch

Initial version

Inhaltsverzeichnis

Intro	v
1. Zusammenbau und erste Inbetriebnahme	1
1.1. Zusammenbau der Hardware	1
1.2. Erster Power-Up	3
1.3. BIOS Einstellungen	3
2. Laden der Firmware	5
2.1. The "Windows Way"	5
2.2. The "Linux Way"	7
2.3. The "Apple Macintosh Way"	8
3. Der erste Bootvorgang	11
4. Der erste Kontakt	13
4.1. Zugriff über die Serielle Schnittstelle	13
4.2. Die Grundkonfiguration	13
4.2.1. Root Passwort setzen	13
4.2.2. System Konfiguration	14
4.2.3. Netzwerk Konfiguration	14
4.2.4. WLAN Konfiguration	15
4.2.5. NTP Konfiguration	17
4.2.6. SNMP Konfiguration	18
5. Erweiterte Funktionen	19
5.1. Netzwerk Features	19
5.1.1. Netzwerk: BGP Routing Daemon	19
5.1.2. Netzwerk: Web Server	19
5.1.3. Netzwerk: DHCP Server	19
5.1.4. Netzwerk: DNS Server	19
5.1.5. Netzwerk: SSH Server	19
5.2. Serielle Schnittstelle	19
A. Performance	21
A.1. Notwendiges SNR für einen stabilen Link	21
B. Aufsetzen einer Build Umgebung	23
B.1. Der kurze Weg	23
B.2. Der lange Weg	23

Intro

Important information, please read me!

Intro

Kapitel 1. Zusammenbau und erste Inbetriebnahme

Important information, please read me!

1.1. Zusammenbau der Hardware

Das folgende Material brauchen wir für einen ALIX HAMNET Router:

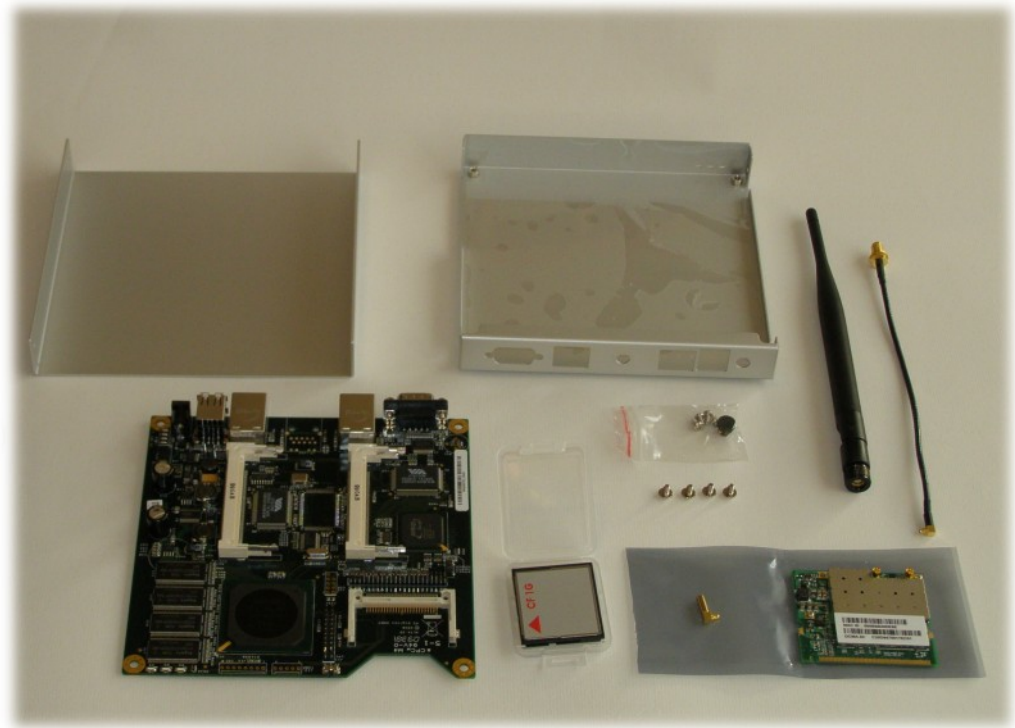


Abbildung 1-1. Inventar eines ALIX HAMNET Routers

- 1 Mainboard ALIX 2d2
- 1 passendes Gehäuse mit Schrauben
- 1 Compact Flash Memory Karte 1GB
- 1 WLAN Modul Atheros DCMA-82
- 1 Abschlusswiderstand
- 1 Anschlusskabel WLAN Modul auf SMA
- 1 Dual-Band Antenne (2.4GHz/5GHz)

Für eine Installation auf einem Höhenstandort wird sicher eine Richtantenne zu Einsatz kommen. Dieses Dokument beschreibt der Einfachheit halber die "Tischversion" eines ALIX Routers.

Achtung, ESD sensitive Komponenten! Um Schäden an den Komponenten zu vermeiden sind ESD Schutzmassnahmen zu treffen.

Als erstes wird die (noch leere) Flash Karte in den dafür vorgesehenen Slot gesteckt, dann wird das ALIX Board in das Gehäuseunterteil geschraubt. Dazu sind zuerst die Arretierschrauben der DSUB-9 Buchse zu entfernen und nach der Montage wieder anzuschrauben.

Das mini-PCI WLAN Modul wird im Slot in der mitte des Boards installiert. Darunter befinden sich weniger Integrierte Komponenten, die Gefahr eines Wärmestaus ist kleiner.

Kapitel 1. Zusammenbau und erste Inbetriebnahme

Achtung: Den 50 Ohm Abschlusswiderstand auf dem AUX Antennenstecker nicht vergessen!



Abbildung 1-2. Innenansicht nach dem Zusammenbau



Abbildung 1-3. Ansicht von Hinten

1.2. Erster Power-Up

Für den ersten Power-up muss die serielle Schnittstelle der ALIX mit dem PC verbunden werden. Auf dem PC wird ein Terminalprogramm (z.B. PuTTY) mit den Einstellungen 38400 Baud, 8/1/n benötigt. Nach dem Einschalten der Speisung sollte in etwa folgende Bootmeldung erscheinen:

```
PC Engines ALIX.2 v0.99h
640 KB Base Memory
261120 KB Extended Memory
Waiting for HDD ...

01F0 Master 044A CF 1GB
Phys C/H/S 1966/16/63 Log C/H/S 983/32/63
[...]
```

Natürlich schlägt der Bootvorgang fehl da noch kein Betriebssystem auf der Compact-Flash Karte installiert ist.

1.3. BIOS Einstellungen

Um zu den BIOS Einstellungen zu gelangen muss die Taste 'S' gedrückt werden bevor der Bootvorgang beginnt. Sobald der RAM-Test läuft, die Taste 'S' wiederholt drücken bis die BIOS Einstellungen erscheinen.

```
PC Engines ALIX.2 v0.99h
640 KB Base Memory
261120 KB Extended Memory
Waiting for HDD ...

01F0 Master 044A CF 1GB
Phys C/H/S 1966/16/63 Log C/H/S 983/32/63

BIOS setup:

(9) 9600 baud (2) 19200 baud *3* 38400 baud (5) 57600 baud (1) 115200 baud
```

Kapitel 1. Zusammenbau und erste Inbetriebnahme

```
*C* CHS mode (L) LBA mode *W* HDD wait (V) HDD slave (U) UDMA enable
(M) MFGPT workaround
(P) late PCI init
*R* Serial console enable
*E* PXE boot enable
(X) Xmodem upload
(Q) Quit
```

Die BIOS Einstellungen werden wie folgt gesetzt:

- (3) Serielle Baudrate: 38400
- (C) "CHS mode" eingeschaltet
- (W) "HDD wait" eingeschaltet
- (R) "Serial Console Enable" eingeschaltet
- (E) "PXE boot enable" eingeschaltet

Diese Einstellung erlaubt uns die ALIX bei Bedarf über das Netzwerk zu booten (PXE Boot, mittels DHCP und TFTP).

Kapitel 2. Laden der Firmware

Der hier beschriebene Weg ist einer von vielen Möglichkeiten das OpenWRT Image auf die Flash Karte zu brennen. Eine andere Möglichkeit ist das direkte Beschreiben der Karte mit dem Linux Tool 'dd'.

2.1. The "Windows Way"

Benötigt werden:

- DHCP und TFTP server. Bewährt hat sich TFTP32 der hier zu finden ist: http://tftpd32.jounin.net/tftpd32_download.html
- Die notwendigen Files für PXE sind hier in einem ZIP Archive zu finden: http://hamnet.tuxworld.ch/download/images_Alix/Bootstrap_PXE/pxe.zip
- Das zu flashende OpenWRT Image: http://hamnet.tuxworld.ch/download/images_Alix/xxxx/openwrt-x86-squashfs.image

Am Ende muss folgende Directory Struktur existieren:

```
C:\Data\ALIX\openwrt-x86-squashfs.image
C:\Data\ALIX\pxe\
C:\Data\ALIX\pxe\openwrt-x86-ext2.fs.gz
C:\Data\ALIX\pxe\openwrt-x86-vmlinuz
C:\Data\ALIX\pxe\pxelinux.0
C:\Data\ALIX\pxe\pxelinux.cfg\
C:\Data\ALIX\pxe\pxelinux.cfg\default
```

Nach der Installation von TFTP32 muss dieser noch konfiguriert werden. Die Einstellungen sind gemäss den folgenden Darstellungen vorzunehmen.

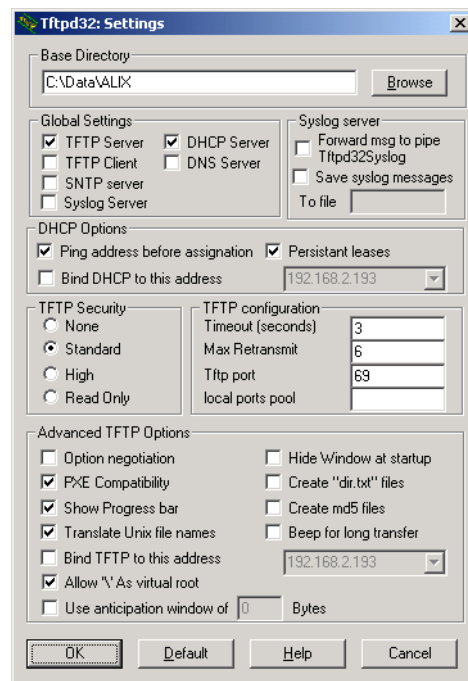


Abbildung 2-1. Allgemeine Einstellungen am TFTP32

Wichtig ist der korrekte Pfad des ALIX Verzeichnisses.

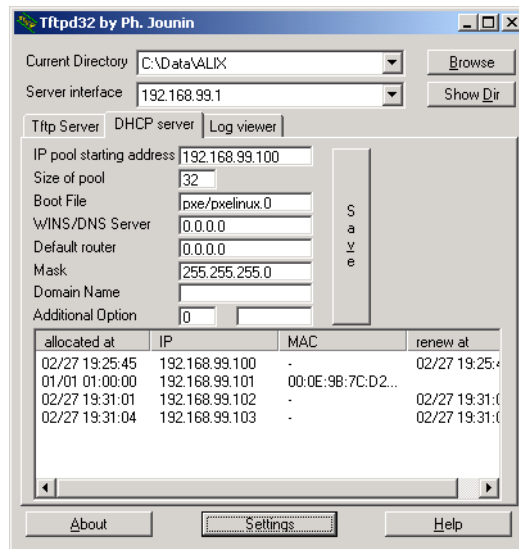


Abbildung 2-2. DHCP Einstellungen am TFTP32

Für den Der DHCP Server verwenden wir das Netz 192.168.99.0/24. Es werden IP Adressen beginnend von 192.168.99.100 verteilt. Das Ethernet LAN Interface von Windows wird mit einer statischen IP Adresse konfiguriert:

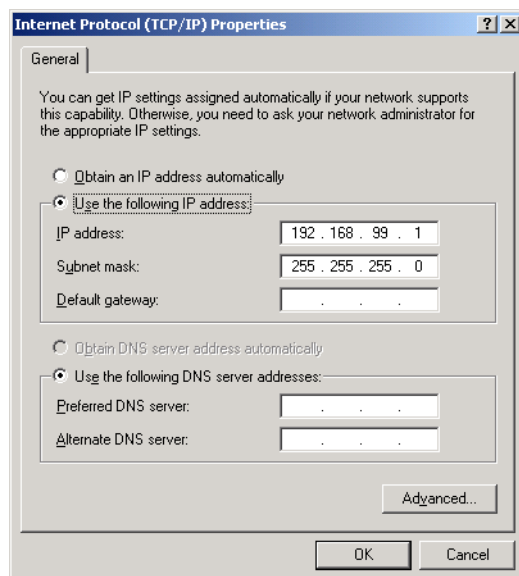


Abbildung 2-3. Windows TCP/IP Konfiguration

Der PC wird mit der ersten Ethernet Schnittstelle (das ist diejenige neben den USB Ports, auf der rechten Seite von hinten gesehen) mit dem PC verbunden. Ebenso wird sie serielle Schnittstelle mit dem PC verbunden und das Terminalprogramm (38400, 8/1/n) gestartet. TFTP32 wird gestartet und danach die Speisung der ALIX eingeschaltet. Auf der seriellen Konsole ist nun der PXE Bootvorgang zu sehen:

```

40 KB Base Memory
261120 KB Extended Memory
Waiting for HDD ...

01F0 Master 044A CF 1GB
Phys C/H/S 1966/16/63 Log C/H/S 983/32/63

Intel UNDI, PXE-2.0 (build 082)
    
```


2.2. The "Linux Way"

Gleiches Vorgehen wie unter Windows. Diese Prozedur zu erarbeiten wird dem geneigten Linux Benutzer als kleine Aufgabe überlassen - sprich: die Dokumentation hierfür folgt später. Falls sich jemand die Mühe macht dies zu dokumentieren: "hier könnte Ihre Anleitung stehen".

2.3. The "Apple Macintosh Way"

Beschreibung für Max OS X 10.6.6. Dieses Kapitel entstand durch einen Beitrag von Konrad Schnetzler, HB9WAD.

Benötigt werden:

- DHCP und TFTP server.
- Die notwendigen Files für PXE sind hier in einem ZIP Archive zu finden: http://hamnet.tuxworld.ch/download/images_Alix/Bootstrap_PXE/pxe.zip
- Das zu flashende OpenWRT Image: http://hamnet.tuxworld.ch/download/images_Alix/xxxx/openwrt-x86-squashfs.image
- USB-RS232 Adapter, Keyspan USA-19HS
- Terminal Programm, Z-Term v1.1beta7 (<http://homepage.mac.com/dalverson/zterm/>)

Da mein MacBook keine RS232 Schnittstelle hat, verwende ich einen Keyspan COM-RS232-Adapter. Dieser wird mit Mac OS X Treiber verkauft. Es funktionieren aber durchaus andere günstigere Adapter mit den gängigen Prolific oder FTDI Chips. Als Terminalprogramm setze ich das betagte Z-Term ein. Es ist nur als PowerPC Binary vorhanden und erscheint noch im alten Mac OS 9 Style. Und trotzdem läuft es noch unter dem neusten 64 bit Mac OS X? Alternativ kann z.B. das aktuelle, aber teure ZOC/Pro 6 verwendet werden. Es läuft ohne Lizenz problemlos im 30-Tage Testmodus.

Mein ALIX.3D2 ist standardmässig auf 38400 Baud, no parity, 8 databits eingestellt. Die BIOS Einstellungen via serielle Schnittstelle werden wie in Kapitel 1.3 beschrieben angepasst.

Bevor es losgehen kann müssen die Entwicklerwerkzeuge Xcode von Apple installiert werden. Danach muss der Sourcecode des DHCP Projektes von <http://www.isc.org/> heruntergeladen werden.

Das Bauen des DHCP Projektes erfolgt mit:

```
sudo -i
cd /opt/local/src
cp ~/myuser/Downloads/dhcp-4.1-ESV-R1.tar .
tar -xvf dhcp-4.1-ESV-R1.tar
cd dhcp-4.1-ESV-R1
./configure
make
```

So, wir installieren nur die Teile für den Server. Andernfalls überschreiben wir möglicherweise Dateien des Apple DHCP Clients.

```
cd ./server
make install
cd ../common
make install
cd ../dhcpctl
make install
cd ../omapip
```

Es folgt die minimale Konfiguration des DHCP Daemon:

```
cd /etc
vi dhcpd.conf
```

Das Konfigurationsfile sollte wie folgt aussehen (die MAC Adresse des ALIX Boards muss natürlich angepasst werden):

```
## /etc/dhcpd.conf file
#
# global options
ddns-update-style ad-hoc;
option domain-name-servers 192.168.1.1, 204.127.202.19;
option domain-name "home.com";

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.101 192.168.1.109;
    default-lease-time 300000;
    max-lease-time 350000;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
}

group {
    host alix {
        hardware ethernet 00:0D:B9:21:5A:3C;
        fixed-address 192.168.1.201;
    }
}
```

Danach das File für die DHCP Leases anlegen:

```
touch /var/db/dhcpd.leases
```

Starten des DHCP Daemon für das Interface /dev/en0:

```
cd /usr/local/sbin
./dhcpd en0
```

Der nächste Schritt ist das Einrichten des TFTP Servers. Mac OS X 10.6 beinhaltet bereits einen TFTP Server. Bei <http://ww2.unime.it/flr/tftpserver/> gibt es ein Tool um den TFTP Server komfortabel per GUI zu konfigurieren.

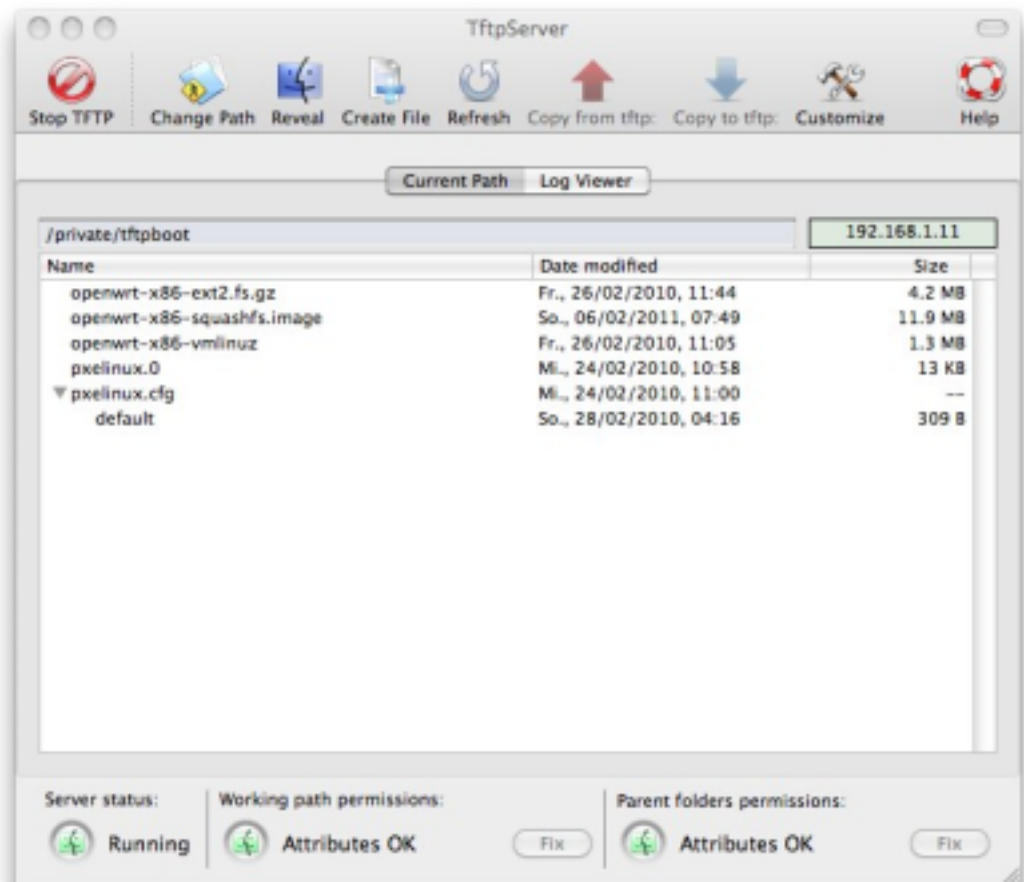


Abbildung 2-4. Konfigurationstool für den TFTP Server

Die Filestruktur innerhalb des TFTP Root Verzeichnisses (/private/tftpboot/) muss wie folgt aussehen:

```
.../openwrt-x86-squashfs.image
.../pxe/
.../pxe/openwrt-x86-ext2.fs.gz
.../pxe/openwrt-x86-vmlinuz
.../pxe/pxelinux.0
.../pxe/pxelinux.cfg/
.../pxe/pxelinux.cfg/default
```

Danach wird der Mac entweder direkt über ein Patchkabel oder über einen Switch mit der ALIX verbunden. Je nach Anforderungen wird das ALIX Board mit separater Speisung versorgt oder mittels passivem PoE Injector über das Netzkabel gespiesen.

Danach wird die ALIX über das Netz gebootet. Im PXE Boot Loader wird durch Eingabe des Buchstaben 'b' das Laden eines initialen OpenWRT gestartet. Danach muss das zu flashende OpenWRT Image ins /tmp Verzeichnis der ALIX kopiert werden (z.B. mit tftp) und mit 'sysupgrade' wird das Compact Flash geschrieben.

```
cd /tmp
tftp -g -r openwrt-x86-squashfs.image 192.168.1.11
sysupgrade -n openwrt-x86-squashfs.image
```


Kapitel 3. Der erste Bootvorgang

Hier folgen einige wissenswerte Fakten zur initialen Konfiguration.

Die Standardkonfiguration:

- Ethernet: Beide Ethernet Ports sind gebridget
- Ethernet: ALIX sucht eine IP Adresse mit DHCP
- Wireless: Ist ausgeschaltet
- Zugriff: Telnet ist aktiv bis ein Root Password gesetzt wird, danach SSH.
- NTP: sucht Timeserver in pool.ntp.org. Anpassen.

Beim allerersten Bootvorgang wird das Flash Filesystem eingerichtet. Es besteht aus einem Read-Only Layer (squashfs) und einem darüberliegenden Read-Write Layer (jffs2). Nach der Installation befinden sich die Files und Standardkonfiguration sich im Read-Only Layer. Nachträglich installierte Pakete oder Modifikationen der Konfiguration werden im darüberliegenden Read-Write Layer gespeichert.

Die erste Konfiguration wird entweder via die serielle Schnittstelle gemacht, oder über eine TELNET Verbindung über das Ethernet. Dazu muss allerdings der DHCP Server auf dem PC wieder gestartet werden. Damit ALIX beim nächste Reset nicht wieder das PXE Image bootet kann im TFTP32 das Feld "Boot Image" (DHCP Server Tab) gelöscht werden. Es kann auch im BIOS der ALIX der PXE Boot ausgeschaltet werden, aber Achtung: ALIX lässt sich dann nicht mehr über das Netz booten ohne das PXE Boot Feature vorher wieder im BIOS einzuschalten! Auf dem Berg ist es in der Regel wünschenswert am "sicheren" Ende des Ethernet Kabels mit dem Laptop zu hantieren und nicht in 37 Metern Höhe auf einem schwankenden Masten.

Kapitel 4. Der erste Kontakt

Nach dem flashen und einem Reset (Power-cycle) kann der Router über das Netzwerk (LAN Interface) und über die serielle Schnittstelle erreicht werden. In der Standardkonfiguration sind die beiden Ethernet Schnittstellen als Switch zusammengeschaltet und versuchen beim booten mit DHCP ein IP Adresse zu holen.

Solange kein Passwort gesetzt wurde, kann mit TELNET eine Shell geöffnet werden. Eine der ersten Aktionen sollte das setzen des Root Passwortes sein. Danach ist TELNET nicht mehr aktiv und der Zugriff muss über SSH erfolgen. SSH Clients gibt es für alle Betriebssysteme - bei Linux ist es "all inclusive", bei Windows ist PuTTY der wohl bekannteste Vertreter.

4.1. Zugriff über die Serielle Schnittstelle

Obwohl der Router auch mit TELNET (SSH) über die LAN Schnittstelle erreichbar wäre konzentrieren wir uns hier auf den seriellen Zugriff um die Grundkonfiguration durchzuführen.

Die Schnittstellenparameter sind 38400,8,1,N. Über ein Nullmodem Kabel wird der PC mit einem Terminalprogramm an den seriellen Port der ALIX angeschlossen.

Hinweis: PuTTY beherrscht auch serielle Kommunikation und ist durch seine gute VT100 Emulation ein excellentes Terminalprogramm.

Also serielle Verbindung herstellen und den Router booten. Der ganze Bootvorgang kann an der seriellen Konsole verfolgt werden. Beim allerersten booten nach dem brennen des OpenWRT Images ist etwas Geduld gefragt, das Einrichten des Filesystems (R/W Layer) dauert eine Minute oder zwei.

Wenn der Bootvorgang abgeschlossen ist kann mit <RETURN> eine Shell gestartet werden. Ein Passwort is an der seriellen Schnittstelle nicht notwendig.

4.2. Die Grundkonfiguration

Aufgrund einiger HAMNET spezifischer Einstellungen sowie der Vielfalt an Paketen gibt es zur Zeit keine Grafische (Web) Oberfläche für die Routerkonfiguration. Vielleicht wird es in der Zukunft einmal ein Web-basiertes Konfigurationstool für die Grundfunktionen geben.

Die folgende Liste zeigt die notwendigen 6 Schritte um eine vernünftige Grundkonfiguration zu erstellen:

- Root Passwort setzen
- /etc/config/system
- /etc/config/network
- /etc/config/wireless
- /etc/ntp.conf
- /etc/config/snmpd

4.2.1. Root Passwort setzen

Das Root Password wird mittels 'passwd' gesetzt.

```
root@alix-3:~# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
root@alix-3:~#
```

Es kann jederzeit wieder geändert werden. Nachdem das Root Passwort gesetzt wurde kann nicht mehr mit TELNET auf den Router zugegriffen werden sondern es muss SSH verwendet werden.

4.2.2. System Konfiguration

Definiert unter anderem den lokalen Hostname und die Zeitzone in der das System läuft (wichtig für NTP Zeitsynchronisierung). Zeitzone wird wohl in der Regel UTC sein.

/etc/config/system:

```
config system
    option hostname HB9AM-3
    option timezone UTC

config rdate
    list server ac-ntp0.net.cmu.edu
    list server ptbtime1.ptb.de
    list server ac-ntp1.net.cmu.edu
    list server ntp.xs4all.nl
    list server ptbtime2.ptb.de
    list server cudns.cit.cornell.edu
    list server ptbtime3.ptb.de
```

Die Liste der rdate Server wird verwendet um beim booten einmal die genaue Zeit zu holen. Dies wird nur getan wenn es eine Default Route gibt (0.0.0.0/0). *Hinweis:* Der ganze rdate Konfigurationsblock sollte gelöscht werden falls er existiert. Bei HAMNET wird die Zeitsynchronisierung durch NTP übernommen.

Das finale /etc/config/system sieht also folgendermassen aus:

```
config system
    option hostname HB9AM-3
    option timezone UTC
```

4.2.3. Netzwerk Konfiguration

Für jeder Netzwerkinterface im System werden hier die IP Adressen und sonstige Einstellungen definiert.

/etc/config/network:

```
config interface loopback
    option ifname lo
    option proto static
    option ipaddr 127.0.0.1
    option netmask 255.0.0.0

config interface lan
    option type bridge
    option ifname "eth0 eth1"
    option proto static
    option ipaddr 192.168.2.103
    option netmask 255.255.255.0
    option gateway 192.168.2.1

config interface wlan
    option proto static
    option ipaddr 10.0.0.3
    option netmask 255.255.255.0

config interface wlanmonitor
    option proto none
```

config interface loopback

Das Loopback Interface gibt es immer. Hier sind keine Änderungen zu machen.

```
config interface lan
```

Standardmässig sind die ersten beiden Ethernet Ports (eth0 und eth1) zusammengebridged. D.h. die beiden Ethernet Ports verhalten sich wie ein Switch. Die IP Adresse und Netzmaske müssen natürlich angepasst werden.

```
config interface wlan
```

IP Adresse und Netzmaske des WLAN Interfaces. Ist natürlich auch den lokalen Gegebenheiten anzupassen.

```
config interface wlanmonitor
```

Das letzte Interface "wlanmonitor" ist ein ein zweites (virtuelles) Interface zur WLAN Karte, das im Monitor Mode betrieben wird. Damit lässt sich der WLAN Verkehr auf der Ebene von 802.11 (also inclusive Baken, Associate Requests, etc.) mitschneiden. Das Tool der Wahl hierfür heisst TCPDUMP. Der Empfangspegel und Noisefloor wird auch mitgeschnitten.

Hinweis: Für eine genaue Beschreibung der einzelnen Konfigurationsoptionen sei auf die OpenWRT Dokumentation verwiesen.

4.2.4. WLAN Konfiguration

Definiert die Einstellungen des Wireless Radios.

Achtung: Hier sicherstellen, dass "option type" auf mac80211_ath5k gesetzt ist. Dies ist notwendig damit die erweiterten WLAN Features (keine Leistungsbeschränkung, 10/5MHz Kanäle, ...) benutzt werden können! Das wifi-device muss als phy<x> benannt sein, nicht radio<x>, wie es bei neuen OpwnWRT Versionen der Fall sein kann.

```
/etc/config/wireless:
```

```
config wifi-device phy0
    option type mac80211_ath5k
    option channel 166
    option macaddr 00:0b:6b:22:a1:2c
    option txpower 4
    option country XA
    option hwmode 11a
    option channelbw 10
    option ant_mode 1
    option distance 10000

config wifi-iface
    option device phy0
    option network wlan
    option mode sta
    #option macaddr 00:11:22:33:44:55
    option ssid HAMNET-HB9XAR
    option encryption none

config wifi-iface
    option device phy0
    option network wlanmonitor
    option mode monitor
```

Die erste Sektion beschreibt die gemeinsamen Einstellungen, für alle virtuellen WLAN Interfaces gemeinsam sind. Dazu gehören die Kanalnummer, Sendeleistung, Bandbreite, Distanz, etc.

```
option type mac80211_ath5k
```

Selektiert den modifizierten WLAN Treiber mit den HAMNET Erweiterungen. Diese Einstellung muss nach dem flashen und ersten booten einmal angepasst werden und sollte danach nicht verändert werden.

```
option channel 166
```

Die WLAN Kanalnummer. Eine Liste der von der WLAN Karte unterstützen Kanäle kann wie folgt angezeigt werden:

```
# iw phy0 info
```

```
option macaddr 00:0b:6b:22:a1:2c
```

Wird beim ersten booten automatisch erstellt. Damit wird die WLAN Karte anhand ihrer MAC Adresse referenziert. Falls die WLAN Karte gewechselt wird (andere MAC Adresse) muss sie hier angepasst werden.

```
option txpower 4
```

Sendeleistung in dBm. 10 dBm entsprechen 10mW, 20 dBm entsprechen 100mW. Die DCMA-82 Karte von Atheros liefert (abhängig von der Bitrate) bis über 23dBm.

```
option country XA
```

Dieser spezielle Ländercode 'XA' beinhaltet das 5GHz Hamradio Band ohne künstliche Leistungsbeschränkung. Ausserhalb des Hamradio Bandes gelten die HB9 Beschränkungen für 802.11/a/b/g.

```
option hwmode 11a
```

Die WLAN Karte soll nur in 802.11a (5GHz) Betrieben werden.

```
option channelbw 10
```

Die Kanal Bandbreite. Kann 20 (default), 10 oder 5 MHz sein.

```
option ant_mode 1
```

Antennen-Mode. Die Atheros DCMA-82 Karte hat 2 Anschlüsse für Antennen (Main und Aux). Wenn mehrer Antennen angeschlossen sind lässt sich z.b. Diversity Empfang realisieren.

Tabelle 4-1. Atheros ant_mode

ant_mode	Funktion	Beschreibung
0	DEFAULT	default antenna setup
1	FIXED_A	only antenna A is present
2	FIXED_B	only antenna B is present
3	SINGLE_AP	sta locked on a single ap
4	SECTOR_AP	AP with tx antenna set on tx desc
5	SECTOR_STA	STA with tx antenna set on tx desc
6	DEBUG	Debug mode -A -> Rx, B-> Tx-

Im HAMNET Regelfall (Links) wird wohl jeweils eine Antenne am Main Antennenport angeschlossen sein, also ant_mode=1. Bei 5GHz Userzugängen kann man sich durchaus andere Szenarien vorstellen.

```
option distance 10000
```

Distanz zum Linkpartner. Diese Zahl bestimmt das Timing auf dem WLAN Link. Daraus abgeleitet werden Slot-Time (bekannt aus der Packet Radio Welt) und ACK/CTS Timeouts. Wird diese Distanz zu klein eingestellt "funktioniert es einfach nicht"; man sieht die Baken der Gegenstelle, es kommt aber keine Association zustande.

Die maximal mögliche Distanz hängt von der Kanalbandbreite ab. Diese Limiten sind durch die Atheros Hardware gegeben.

Tabelle 4-2. Geschätze maximale Distanzen

Bandbreite	Max. Distanz
5	108 km
10	54 km
20	27 km

Die weiteren Sektionen enthalten die Konfiguration für die individuellen (virtuellen) WLAN Interfaces:

```
option device phy0
```

Verweis auf das Parent-Device.

```
option network wlan
```

Zuweisung der WLAN Interfaces in ein Netzwerk. Wird in /etc/config/network referenziert um die IP Adresse zu definieren.

```
option mode sta
```

Betriebsmodus des WLAN Interfaces:

Tabelle 4-3. WLAN Interface Modi

Mode	Beschreibung
ap	Access Point
sta	Station (verbindet sich zu einem AP)
adhoc	Ad-Hoc Netzwerk
monitor	Monitor Mode.

Ein HAMNET Link (Point-to-Point) wird in der Regel auf einer Seite einen AP (Access Point) haben und auf der anderen Seite eine STA (Station). Bei Point-to-Multipoint Links (wenn z.B. zwei Gegenstationen mit derselben Antenne abgedeckt werden) ist der Sternpunkt als AP zu konfigurieren und alle Gegenstationen sind STA.

```
option macaddr 00:11:22:33:44:55
```

Erlaubt es, die MAC Adresse des Interfaces frei zu definieren. Dies ermöglicht das bei DB0FHN beschriebene Verfahren ein Callsign als eine MAC Adresse darzustellen¹.

```
option ssid HAMNET-HB9XAR
```

Die SSID des WLANs.

```
option encryption none
```

Und die Verschlüsselung schalten wir natürlich ab.

4.2.5. NTP Konfiguration

```
/etc/ntp.conf:
```

```
#restrict default nomodify notrap noquery
#restrict default noquery

restrict 127.0.0.1

driftfile /tmp/ntp.drift

server 44.142.64.16 iburst
```

Die vordefinierten NTP Server aus dem ntp.org Pool sind zu entfernen und durch einen (oder mehrere) lokale NTP Server zu ersetzen. Die kann z.B. ein benachbar-

ter ALIX Knoten sein oder ein im HAMNET ansässiger NTP server. In der Standardkonfiguration ist der NTP Server "offen", d.h. andere Knoten können sich darauf synchronisieren.

4.2.6. SNMP Konfiguration

Der integrierte SNMP Daemon muss natürlich auch konfiguriert werden. Mindestens Location, Contact und Description sollten korrekt konfiguriert sein.

/etc/config/snmpd:

```
config system
    option sysLocation      'JN47AI'
    option sysContact       'hb9xar@uska.ch'
    option sysName          'ALIX-3'
#   option sysServices     72
    option sysDescr        'ALIX HAMNET Router'
#   option sysObjectID     '1.2.3.4'
```

Die folgenden Zeilen sind anzufügen:

```
config exec
    option execname temperature
    option prog      /usr/bin/gettemp.sh
    option args
    option miboid    1.3.6.1.3.999.1.1

config pass
    option miboid    1.3.6.1.3.999.2.1
    option prog      /usr/bin/getwlaninfo.sh
    option persist   0
```

Damit sind aktuelle Temperatur und WLAN Daten via SNMP ablesbar. Mit Paketen wie RRDTOOL und Nagios lassen sich dann die Router überwachen und Statistiken anfertigen.

Details zum Mapping der MIBS (im "experimental" Bereich) finden sich in den Kommentaren von /usr/bin/gettemp.sh und /usr/bin/getwlaninfo.sh

Fußnoten

1. <http://db0fhn.efi.fh-nuernberg.de/doku.php?id=projects:wlan:proposal>

Kapitel 5. Erweiterte Funktionen

To be defined.

5.1. Netzwerk Features

To be defined.

5.1.1. Netzwerk: BGP Routing Daemon

To de defined.

5.1.2. Netzwerk: Web Server

To de defined.

5.1.3. Netzwerk: DHCP Server

To de defined.

5.1.4. Netzwerk: DNS Server

To de defined.

5.1.5. Netzwerk: SSH Server

To de defined.

5.2. Serielle Schnittstelle

To de defined.

Kapitel A. Performance

Dieses Kapitel zeigt die Resultate einiger Tests und Performanceuntersuchungen.

Achtung: Der HAMNET WLAN Treiber auf einer ALIX zeigt die Bitrate des Kanals immer bezogen auf einen 20 MHz Kanal an. Bei 10 MHz (Half-rate) und 5 MHz (Quarter-rate) Kanälen sind die angezeigten Werte umzurechnen (also /2 oder /4)!

A.1. Notwendiges SNR für einen stabilen Link

Frage: Was ist das notwendige SNR um eine bestimmte Datenrate auf dem Link zu erhalten? Oder anders gefragt, wieviel RX Pegel brauche ich um einen WLAN Link stabil zu betreiben?

Im Internet finden sich einige Informationen, darunter auch dieser Link hier¹. Die Tabelle zeigt das (nach Meinung von Cisco) minimal notwendige SNR um eine bestimmte Bitrate zu erreichen. Bei zu schlechtem SNR schaltet die WLAN Karte automatisch auf eine tiefere Bitrate.

Tabelle A-1. Bitrate in Funktion des SNR (20 MHz Bandbreite)

Bitrate (11ag)	6 Mbps	9 Mbps	12 Mbps	18 Mbps	24 Mbps	36 Mbps	48 Mbps	54 Mbps
SNR	4 dB	5 dB	7 dB	9 dB	12 dB	16 dB	20 dB	24 dB

Umgerechnet auf 5MHz Bandbreite ergibt das:

Tabelle A-2. Bitrate in Funktion des SNR (5 MHz Bandbreite)

Bitrate (11ag)	1.5 Mbps	2.25 Mbps	3 Mbps	4.5 Mbps	6 Mbps	9 Mbps	12 Mbps	13.5 Mbps
SNR	4 dB	5 dB	7 dB	9 dB	12 dB	16 dB	20 dB	24 dB

Um diese theoretischen Werte zu überprüfen wurden an der Versuchsstrecke DB0TIT-DB0WBD Messungen vorgenommen. Dazu wurde die Ausgangsleistung auf beiden Seiten (DB0TIT Seite als AP und DB0WBD Seite als STA) schrittweise reduziert und den Einfluss auf den Empfangspegel und Bitrate an der Gegenstelle ermittelt.

Ein paar Daten zu der Versuchsstrecke:

A-Ende: DB0TIT (Access Point)
 Z-Ende: DB0WBD (Station)
 Distanz: ca. 17 km
 Bandbreite: 5MHz
 Frequenz: 5.685 GHz (802.11 Kanalnummer 137)

Dabei ergibt sich folgende Situation:

Tabelle A-3. Bitrate in Funktion des SNR (5 MHz Bandbreite)

TX Power (AP)	RX Level (STA)	TX Rate (AP)	SNR (STA)	TX Power (STA)	RX Level (AP)	TX Rate (STA)	SNR (AP)
17 dBm	-82 dBm	12 Mbps	25 dB	17 dBm	-80 dBm	13.5 Mbps	27 dB

TX Power (AP)	RX Level (STA)	TX Rate (AP)	SNR (STA)	TX Power (STA)	RX Level (AP)	TX Rate (STA)	SNR (AP)
13 dBm	-87 dBm	9 Mbps	20 dB	13 dBm	-84 dBm	13.5 Mbps	23 dB
10 dBm	-95 dBm	9 Mbps	12 dB	10 dBm	-90 dBm	12 Mbps	17 dB
7 dBm	-97 dBm	4.5 Mbps	10 dB	7 dBm	-94 dBm	9 Mbps	13 dB
6 dBm	-99 dBm	3 Mbps	8 dB	6 dBm	-96 dBm	4.5 Mbps	11 dB
4 dBm	-103 dBm	1.5 Mbps	4 dB	4 dBm	-100 dBm	2.25 Mbps	7 dB
3 dBm	kein Link	-	-	3 dBm	kein Link	-	-

Der Noise-Floor liegt bei der genannten Versuchsstrecke bei ca -107dBm (der Wert, den die WLAN-Karte als Noise-Floor ermittelt). Diese, in der realen Welt ermittelten, SNR Werte Decken sich in etwa mit den Erwartungen. Interessant ist auch, dass mit -103 dBm Empfangspegel (4 dB SNR) die WLAN Strecke bereits läuft - auf der niedrigsten Bitrate.

Ziel muss also ein SNR > 24 dB sein (+ Reserve) um den Link mit voller Geschwindigkeit betreiben zu können. Wenn sich das SNR verschlechtert wird die Bitrate auf dem Kanal *adaptiv* angepasst, d.h. der Link läuft auch bei massiv schlechterem Signal - allerdings mit reduzierter Bandbreite.

Die Praxis muss zeigen ob dies genügend Reserven gibt um die Umgebungseinflüsse (Regen, Nebel, Schnee, ...) zu kompensieren.

Fußnoten

1. http://www.cisco.com/en/US/tech/tk722/tk809/technologies_q_and_a_item09186a00805e9a96.s

Kapitel B. Aufsetzen einer Build Umgebung

Dieses Kapitel zeigt die notwendigen Schritte um eine OpenWRT Build-Umgebung aufzusetzen, die notwendigen HAMNet Patches einzuspielen und eigene Images zu bauen.

Der Autor selbst verwendet eine CentOS 5 Linux Distribution; die folgenden Schritte wurden also damit getestet. Natürlich wird die (mehr oder wenige) identische Prozedur auch auf anderen Linux Distributionen funktionieren wenn die notwendigen Pakete installiert sind.

Achtung: Die folgenden Softwarepakete werden benötigt und müssen installiert sein:

- SVN Client
- wget
- GCC Umgebung

B.1. Der kurze Weg

Der schnelle Weg um eine "HAMNet OpenWRT" Build Umgebung zu erstellen geht über ein kleines Script welches alle notwendigen Schritte ausführt: quick_start.sh¹.

Das Script wird in einem leeren Directory gespeichert und erhält Execute Permissions (chmod +x quick_start.sh).

Achtung: Bitte zuerst im Script kontrollieren ob die korrekte OWRT Version und HAMNET Patches heruntergeladen werden. Der OWRT_TAG und HAMNET_TAG müssen unter Umständen angepasst werden.

```
# Script herunterladen:
$ wget http://hamnet.tuxworld.ch/download/development/quick_start.sh

# Execute Permissions
$ chmod +x quick_start.sh

# und ausführen
$ ./quick_start.sh
```

Der Prozess dauert einige Minuten und muss ohne Fehler durchlaufen.

B.2. Der lange Weg

Der lange Weg mach die gleichen Schritte mit dem Ziel die OpenWRT Build Umgebung etwas näherzubringen. Es handelt sich dabei um 11 Schritte. Wir betrachten hier konkret den OWRT Release "backfire 10.03".

Bevor es losgeht wird zuerst ein leeres Directory angelegt:

```
$ mkdir OpenWRT
$ cd OpenWRT
```

Schritt 1: OpenWRT Code aus dem SVN Repository holen:

```
$ svn co svn://svn.openwrt.org/openwrt/tags/backfire_10.03/
```

Schritt 2: HAMNet Patches holen:

Ein Directory für die Patches anlegen und alle HAMNET Patches² in dieses Directory kopieren.

Kapitel B. Aufsetzen einer Build Umgebung

```
$ mkdir patches
```

Die folgenden Files werden benötigt:

- 9999-xar_wireless.patch
- HAMNET_OWRT.key.pub.pem
- OWRT_feeds_packages.patch
- OWRT_trunk.patch
- alix_config_hamnet_1003
- package_hamnet.tgz
- regulatory.bin
- feeds.conf

Schritt 3: Den OWRT_trunk.patch einspielen:

```
$ cd backfire_10.03
$ patch -p0 < ../patches/OWRT_trunk.patch
```

Schritt 4: Den Wireless Patch kopieren:

```
$ cd backfire_10.03
$ cp ../patches/9999-xar_wireless.patch package/mac80211/patches/
```

Schritt 5: Die neue Wireless RegDomain kopieren:

```
$ cd backfire_10.03
$ cp ../patches/HAMNET_OWRT.key.pub.pem package/crda/files/
$ cp ../patches/regulatory.bin package/crda/files/
```

Schritt 6: Das ALIX-HAMNet Config File kopieren:

```
$ cd backfire_10.03
$ cp ../patches/alix_config_hamnet_1003 .config
```

Schritt 7: Das HAMNet Software Paket hinzufügen:

```
$ cd backfile_10.03/package/
$ tar xvzf ../../patches/package_hamnet.tgz
```

Schritt 8: Das feeds.conf file kopieren:

```
$ cd backfile_10.03/
$ cp ../patches/feeds.conf .
```

Das ist notwendig, damit ein definierter Release der Feeds (Add-on Pakete) referenziert wird. Falls dies nicht gemacht wird, wird immer die neueste Version geladen - was dann keinen konsistenten Zustand garantiert.

Schritt 9: Die optionalen OWRT Pakete hinzufügen:

```
$ cd backfile_10.03/
```

```
$ make package/symlinks
```

Das kann ein paar Minuten dauern.

Schritt 10: Den OWRT_feeds Patch einspielen:

```
$ cd backfire_10.03/feeds/packages/  
$ patch -p0 < ../../../../patches/OWRT_feeds_packages.patch
```

Schritt 11: Jetzt können wir das ganze bauen:

```
$ cd backfire_10.03/  
$ make clean  
$ make world
```

Zuerst wird die ganze GCC Toolchain (Compiler, C Libraries, ...) gebaut und un einem zweiten Schritt schliesslich alle OpenWRT Pakete.

Warum wird eigentlich der Compiler auch gebaut, der ist ja bereits installiert? Grund ist, dass egal auf welchem Linux Release und mit welchen Compilern / Bibliotheken gebaut wird, immer die gleichen Compiler und C Libraries für OpenWRT verwendet werden. Ebenso ermöglicht dieses Konzept das "Cross-comiling", d.h. das bauen von von binärem Code für eine andere Plattform (z.b. gebaut wird auf einem Intel x86 Linux PC, Zielplattform ist aber ARM oder MIPS).

Fußnoten

1. http://hamnet.tuxworld.ch/download/development/quick_start.sh
2. <http://hamnet.tuxworld.ch/download/development/testing/>

