
MESHKIT DOCUMENTATION

Version 0.0.2

Aktualisiert am 26. 10. 2013

Manuel Munz

<http://www.grund-wissen.de>

1	Allgemeine Infos zur Meshkit Freifunk Firmware	1
1.1	Was ist Meshkit?	1
1.2	Features	1
2	Einbinden des Freifunkrouters ins eigene Netzwerk	3
2.1	LAN-Setup	3
2.2	WAN-Setup	3
3	Firmware mit Meshkit generieren lassen	5
3.1	Wichtige Infos zum Router sammeln	5
3.2	IP-Adresse(n) registrieren	6
3.3	Firmware generieren	6
4	Erster Login	9
5	Benutzerhandbuch	11
5.1	Mit dem Router verbinden	11
5.2	Passwort ändern	11
6	Fortgeschrittene Konfiguration	13
6.1	Eigene Dienste anbieten	13
7	Hilf mit bei der Dokumentation	19
7.1	Allgemeine Regeln zur Dokumentation	19
7.2	Installation von Sphinx	19

Allgemeine Infos zur Meshkit Freifunk Firmware

1.1 Was ist Meshkit?

Meshkit ist ein Generator für individualisierte Freifunk-Firmware-Images.

1.2 Features

- Support für Communityprofile (verschiedene Einstellungen für unterschiedliche Communities)
- Erstellen vorkonfigurierter Firmwareimages, die direkt nach dem Flashen im Mesh erreichbar sind.
- Pakete können direkt mit ins Image gebaut werden (spart Platz)
- Eigene Dateien können mit ins Image gebaut werden

Einbinden des Freifunkrouters ins eigene Netzwerk

2.1 LAN-Setup

Der Router ist mit seinem WAN-Port über einen Switch Teil des lokalen Netzwerks. Eigene PCs sind ebenfalls mit diesem Switch verbunden.

2.2 WAN-Setup

Der Router ist mit seinem WAN-Port am Internet (z.B. Modem) angeschlossen. Eigene PCs werden an den LAN-Ports des Routers angeschlossen oder verbinden sich per WLAN.

Firmware mit Meshkit generieren lassen

3.1 Wichtige Infos zum Router sammeln

Um eine passende Firmware für das eigene Routermodell zu generieren muss man zunächst einige Daten zum Router kennen:

- Chipsatz
- Chipsatz des/der WLAN-Interfaces
- Größe des RAM
- Größe des Flash-Speichers

Diese Daten finden man in der [Table of Hardware](#) im OpenWrt Wiki. Zu den meisten Routermodellen gibt es auch eine Detailseite mit weiteren Infos. Diese Seite sollte man ebenfalls zumindest kurz überfliegen. Hier finden sich im Allgemeinen auch Infos zum Flashen und zur Rettung des Routers falls beim Flashen etwas so richtig schief ging.

Beispiel:

Wir wollen die Daten für einen TP-Link WR1043ND zusammentragen. In der [Table of Hardware](#) sehen wir folgendes:

TL-WR1043ND	1-1.8	10.03.1	ar71xx	Atheros AR9132	400	8	32	Atheros AR9100 (integrated)	11b/g/n	Yes x3	5 gig
TL-WR2543ND	1.0	12.09	ar71xx	Atheros AR7242	400	8	64	Atheros AR9380 (onboard)	11a/b/g/n	Yes x3	5 gig
Model	Version	Status	Target(s)	Platform	CPU Speed (MHz)	Flash (MB)	RAM (MB)	Wireless NIC	Wireless Standard	Detachable Antennae	Wireless Power

Daraus können wir entnehmen:

- Der Chipsatz gehört zur AR71XX Familie
- WLAN benutzt Atheros Hardware
- Der RAM ist 32 MB groß
- Flash ist 8 MB groß

3.2 IP-Adresse(n) registrieren

Nun muss noch mindestens eine IP-Adresse aus dem Freifunknetz für den Router registriert werden. Hier hat jede Community ihre eigenen Seiten zur Registrierung, im Allgemeinen kann man sich hier jedoch selbst bedienen und einfach eine noch nicht vergebene Adresse für seinen Knoten reservieren.

Registrierungsseiten in den einzelnen Communities:

- **Augsburg:** Als Benutzer auf <http://augsburg.freifunk.net> einloggen und “Node registrieren”

3.3 Firmware generieren

Das Erstellen von Firmwareimages erfolgt in drei Schritten:

3.3.1 Grundlegende Systemeinstellungen

Öffne die [Meshkit](#) Webseite im Browser. Du siehst nun folgendes:

IndexÜberStatus

Login | Registrieren | Lost password?

Meshkit

Freifunk OpenWrt Imagebuilder

IndexDeutsch

1. Grundlegende Einstellungen

Diese stabile Version des Meshkit baut Attitude Adjustment Images. Bitte beachten, dass Attitude Adjustment nicht mehr auf Geräten läuft, die **nur 16MB Ram haben**. Für diese Geräte bitte die Backfire Images auf [Imagebuilder](#) benutzen.

Community
Target
Expertenmodus
Keine Konfiguration
Email

aachen
ar71xx-generic-attitude_adjustment-36088
☐
☐

SUBMIT

Anhand des Modells wählen

Unten können einige bekannte Routermodelle ausgewählt werden. "Target" auf dieser Seite und "Profil" auf der nächsten Seite werden dadurch automatisch gewählt.
Wenn dein Routermodell nicht angezeigt wird dann wähle "Target" und "Profil" (auf der nächsten Seite) bitte von Hand aus.

TP-Link
WDR3600 | WDR4300 | WR741 | WR1043ND | WR841 | WDR4310
Linksys
WRT54G
Netgear
WGT634U
Siemens
SE505v2
Ubiquiti
Litestation LS-SR71 | Routerstation | Routerstation Pro

Meshkit is a webinterface for the [OpenWrt](#) image generator. Es ermöglicht vorkonfigurierte und angepasste OpenWrt Firmwareimages für Router und Access Points zu erstellen.

Fehler gefunden? Oder Verbesserungsvorschläge/Fragen? Dann besuche die [Über](#) Seite für Kontaktinformationen.
Version: 0.0.1

Erklärung:

1. Login bzw. Registrieren

Hier kann man einen User für Meshkit registrieren. Dadurch wird es möglich, einige Datenfelder beim Generieren neuer Images bereits auszufüllen, z.B. Community, Adresse oder SSH Public Keys. Es ist nicht notwendig einen Benutzer im Meshkit zu registrieren. Wer aber öfter Images generiert dem kann die Registrierung ersparen, bei einigen Feldern immer und immer wieder die selben Daten einzugeben.

2. Geräte Vorauswahl

Im unteren Bereich von Meshkit befinden sich einige Links zu Geräten, die häufig verwendet werden. Klickt man auf einen dieser links, dann wählt Mes-

hkit automatisch das richtige Target (und in Schritt 2 auch das passende Profil) für dieses Gerät. Wenn ein Gerät nicht in dieser Liste steht dann heisst das nicht, dass es nicht unterstützt wird, sondern nur, dass man Target und im nächsten Schritt Profil manuell auswählen muss.

3. Einstellungen

Option	Beschreibung
Community	Wähle deine Community aus. (TODO: Was tun wenn es noch keine gibt?)
Target	Wähle ein passendes Target (siehe Infos zum Router sammeln)
Expertenmodus	Wenn ausgewählt, dann werden im nächsten Schritt wesentlich mehr Optionen zur Konfiguration des Routers angezeigt.
Keine Konfiguration	Erstellt ein Image, es wird aber keine Konfiguration durchgeführt. Dies ist vor allem nützlich um Images zu erhalten, die für sysupgrade (TODO: Seite zu sysupgrade anlegen) verwendet werden können.
Email	Wenn angegeben wird nachdem das Image gebaut wurde eine Mail an diese Adresse geschickt.

4. Absenden

Nachdem alle Einstellungen getätigt wurde klicke auf **Absenden** um zu Schritt 2 des Meshkits zu gelangen, wo weitere Einstellungen vorgenommen werden müssen.

Erster Login

Nach dem Flashen des Routers sollte man sich entweder per Webinterface oder per `ssh` mit dem Node verbinden, um das Passwort zu ändern.

Benutzerhandbuch

5.1 Mit dem Router verbinden

5.1.1 Netzwerkverbindung herstellen

Benötigt: Integrationsszenarien

Es gibt zwei Wege sich mit dem Router zu verbinden, um Status abzufragen oder die Konfiguration zu ändern. Entweder man öffnet in einem Browser die Webseite des Routers oder verbindet sich per ssh.

5.2 Passwort ändern

Setzt voraus: Mit dem Router verbinden

5.2.1 Im Webinterface

Das Passwort kann unter Administration -> System -> Administrator geändert werden.

5.2.2 Mit SSH

`passwd`

und zwei mal ein neues Passwort eingeben.

Fortgeschrittene Konfiguration

Da die Meshkit Firmware auf dem vielseitigen OpenWrt basiert, gibt es viele Möglichkeiten die Software auf dem Router für verschiedenste Einsatzzwecke anzupassen.

6.1 Eigene Dienste anbieten

Es ist möglich, im Freifunknetz eigene Dienste zur Verfügung zu stellen.

6.1.1 Eigene Webseiten

Lokal auf dem Node

Auf dem Freifunkrouter läuft ohnehin schon ein kleiner Webserver (`uhttpd`), der verwendet werden kann um neben dem LuCI Webinterface auch eigene Inhalte per HTTP zugänglich zu machen.

Seiten direkt über `uhttpd` ausliefern

Es gibt die Möglichkeit direkt auf dem Node kleinere Seiten abzulegen, indem man sie im Ordner `/www` unterbringt. Dies erfordert keine Änderung an der Konfiguration des Nodes.

Einfache statische Seite

Beispiel:

```
<html>
  <body>
    Hallo Welt!
  </body>
</html>
```

Den obigen Code als `/www/hallowelt.html` abspeichern und danach im Browser unter der Adresse `http://<nodeip>/hallowelt.html` aufrufen.

CGI-Skripte

Möchte man CGI-Skripte ausführen, dann muss man diese in `/www/cgi-bin` ablegen und ausführbar machen.

Beispiel:

```
#!/bin/sh

echo "Content-type: text/html"
echo ""

echo '
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Hallo Welt</title>
  </head>
  <body>
    Hallo Welt!
  </body>
</html>
'
```

Den obigen Code als `/www/cgi-bin/helloworld.cgi` speichern und ausführbar machen mit:

```
chmod +x /www/cgi-bin/helloworld.cgi
```

Dieses Testscript kann nun im Browser unter `http://<nodeip>/cgi-bin/helloworld.cgi` abgerufen werden.

Das selbe Script in Lua:

```
#!/usr/bin/lua

print("Content-type: text/html")
print("")

print ([[
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Hello World</title>
  </head>
  <body>
    Hallo Welt!
  </body>
]])
```

```
</html>
]])
```

Verzeichnisinhalte

In der Standardkonfiguration zeigt uhttpd auch Verzeichnisinhalte von Verzeichnissen an, die keine index-Datei enthalten.

Beispiel:

```
mkdir /www/verzeichnis
echo "hallo" > /www/verzeichnis/hallo.txt
```

Öffnet man im Browser jetzt <http://<nodeip>/verzeichnis>, dann erhält man folgende Ausgabe:

Index of /verzeichnis/

1. 
modified: Fri, 11 Oct 2013 09:58:14 GMT
directory - 0.00 kbyte
 2. [hallo.txt](#)
modified: Fri, 11 Oct 2013 09:58:31 GMT
text/plain - 0.01 kbyte
-

Auf einem separaten PC/Server

Soll die Webseite von einem anderen PC bzw. Server ausgeliefert werden, der per Netzwerk mit dem Node verbunden ist gibt es mehrere Möglichkeiten, diese Inhalte im Freifunknetz verfügbar zu machen.

Portforwarding

Auf dem Node wird ein Portforwarding eingerichtet, das einkommende Pakete an den Webserver im eigenen Netzwerk weiterleitet. Siehe: Portforwarding (TODO)

HNA

Mit HNA (Host Network Announcement) kann auf dem Node ein verfügbarer IP-Bereich oder auch nur eine einzelne Adresse im Netzwerk bekannt gemacht werden. So ist der Webserver direkt im Mesh erreichbar, muss jedoch selbst nicht OLSR nutzen. Siehe HNA für Rechner im eigenen Netz (TODO)

Server direkt ins OLSR Netz

Indem der Server selbst OLSR “spricht” kann er direkt aus dem Mesh erreichbar sein. Siehe: OLSR im eigenen Netz (TODO)

6.1.2 Dienste ankündigen mit dem Nameservice Plugin

Um eigene Dienste im gesamten Mesh bekannt zu machen kann das OLSR-Nameservice Plugin verwendet werden. Dieses sendet in regelmässigen Abständen Informationen über die lokalen Dienste an alle anderen Nodes im Mesh.

Es können nur Service-Ankündigungen für IPs/Adressen verschickt werden, die entweder local verwendet oder als HNA angekündigt werden.

Ist der Service unter einer DNS-Adresse bekannt, dann kann statt einer IP auch diese Adresse in der URL des Dienstes verwendet werden.

Einrichtung

Ziel: Es soll ein Webserver auf dem lokalen Knoten mit der IP 10.11.12.13 angekündigt werden. Der Webserver läuft auf Port 80.

Einrichtung über LuCI

1. Um zu den Einstellungen für das Nameservice Plugin zu kommen: Gehe zu *Dienste* -> *OLSR* -> *Plugins*



2. Klicke in der Zeile wo `olsrd_nameservice.so.0.3` steht auf “Bearbeiten”

OLSR - Plugins

Plugins

Aktiviert	Bibliothek	
<input checked="" type="checkbox"/>	olsrd_arprefresh.so.0.1	 Bearbeiten
<input checked="" type="checkbox"/>	olsrd_nameservice.so.0.3	 Bearbeiten
<input checked="" type="checkbox"/>	olsrd_bttinfo.so.0.1	 Bearbeiten
<input checked="" type="checkbox"/>	olsrd_watchdog.so.0.1	 Bearbeiten
<input checked="" type="checkbox"/>	olsrd_jsoninfo.so.0.0	 Bearbeiten
<input type="checkbox"/>	olsrd_dyn_gw_plain.so.0.4	 Bearbeiten

 Zurücksetzen  Speichern  Speichern & Anwenden

3. Füge unten aus der Auswahlbox eine Option für “Service” hinzu

OLSR - Plugins

Pluginkonfiguration

Aktivieren ☒

Bibliothek: olsrd_nameservice.so.0.3


suffix:

hosts_file:

latlon_file:




sigchup_pid_file:

services_file:

-- Zusätzliches Feld --  Hinzufügen

-- Zusätzliches Feld --

- name
- hosts
- add_hosts
- dns_server
- resolv_file
- interval
- timeout
- lat
- lon
- latlon_infile
- name_change_script
- service**
- services_change_script
- mac
- macs_file
- macs_change_script

 Zurücksetzen  Speichern  Speichern & Anwenden

4. Plugin konfigurieren

OLSR - Plugins

Pluginkonfiguration

Aktivieren	<input checked="" type="checkbox"/>
Bibliothek:	olsrd_nameservice.so.0.3
suffix	.ffa
hosts_file	/var/etc/hosts.olsr
latlon_file	/var/run/latlon.js
sighup_pid_file	/var/run/dnsmasq.pid
service	http://10.11.0.8:80 tcp Mein Router
services_file	/var/run/services_olsr

-- Zusätzliches Feld --

Einrichtung mithilfe der Konsole

```
uci add_list olsrd.olsrd_nameservice.service="http://10.11.0.8:80|tcp|Mein Router"
uci commit olsrd
/etc/init.d/olsrd restart
```

Es können auch mehrere Dienste angekündigt werden:

```
uci add_list olsrd.olsrd_nameservice.service="http://10.11.0.8:80|tcp|Mein Router"
uci add_list olsrd.olsrd_nameservice.service="ftp://10.11.0.8:21|tcp|Mein FTP Server"
uci commit olsrd
/etc/init.d/olsrd restart
```

Erklärung des Service Strings

Der Aufbau des erwarteten Strings als Option für Service ist recht einfach:

<url>:<port>|<Protokoll (tcp oder udp)>|<Beschreibung des Dienstes>

Port darf nicht weggelassen werden!

Ergebnis

Ist alles korrekt eingerichtet dann erscheint im öffentlichen Teil des Webinterfaces auf allen Routern im Mesh unter “Dienste” nach kurzer Zeit die eben eingerichtete Ankündigung für “Mein Router”.

Dienste			
Internal services			
Url	Protokoll	Quelle	
Mein Router	tcp	my own service	
Augsburger Blog Aggregator	tcp	10.11.10.29	
Bookmarks von soma. Wer will kann da auch nen Account haben.	tcp	10.11.10.10	
Internes Wiki	tcp	10.11.10.10	

Hilf mit bei der Dokumentation

Eine Dokumentation zu schreiben ist jede Menge Arbeit. Hilfe dabei ist daher gerne gesehen.

7.1 Allgemeine Regeln zur Dokumentation

- Texte sinnvoll durch Überschriften gliedern
- Bilder: Nur JPG oder PNG verwenden
- Wo möglich verweise auf andere Stellen in der Dokumentation
- Erkläre möglichst gut und dabei so kurz wie möglich

Zur Erstellung der Dokumentation wird [Sphinx](#) genutzt.

7.2 Installation von Sphinx

Viele Linux Distributionen bieten Sphinx als Paket an, unter Debian kann es installiert werden mit:

```
aptitude install python-sphinx python-pygments
```

Desweiteren benötigt diese Dokumentation ein angepasstes Theme, das mit folgendem Befehl installiert werden kann:

```
easy_install sphinxjp.themes.basicstrap
```

oder alternativ mit

```
pip install sphinxjp.themes.basicstrap
```