

1. OpenWRT auf RB532

Für jede Linux Distribution sind Bootloader, Kernel und Filesystem notwendig. Ebenso im Embedded-Bereich, nur in minimierter Version. Punkt 1.1. und 1.2. beschreiben die Vorgehensweise für die Installation dieser drei Embedded Linux Betriebssystem Komponenten. Punkt 1.3 beschreibt die Implementierung bzw. Nachrüstung fehlender Pakete für RBTemp.

1.1 Vorbereitung Entwicklungsrechner

Ein Intel Pentium 4 (1500 MHz, 384 MB RAM) Standrechner mit Linux Debian 4.0 Betriebssystem fungiert als Entwicklungsrechner (EWR). Zwischen dem EWR und dem RB532 wird zum einen die RS-232 und zum anderen die Ethernet-Verbindung für die Installation der Software am Board verwendet.

1.1.1 Voreinstellungen

Falls noch nicht vorhanden, müssen zusätzliche, aufgelistete Debian-Pakete am EWR nachinstalliert werden.

```
sudo apt-get install subversion build-essential binutils flex bison
autoconf gettext texinfo sharutils subversion libncurses5-dev zlib1g-dev
rsync gawk unzip screen mc rsync tcpdump net-tools
```

DHCP Server

```
root:/# apt-get install dhcp3-server
```

Anpassung des DHCP-Servers am Entwicklungsrechner:

```
root:/# nano /etc/dhcp3/dhcp.conf
```

Interface-Angabe für DHCP Server

```
root:/# nano /etc/default/dhcp3-server
```

TFTP Server

```
root:/# apt-get install tftpd
```

```
root:/# nano /etc/inetd.conf
```

File-Ordner für den TFTP-Datenaustausch erstellt:

```
root:/# mkdir -p tftpboot
```

Nach Konfiguration müssen der DHCP und TFTP Server neu gestartet werden.

1.1.2 Entwicklungsumgebung

Image-Builder

Für die Erstellung eines neuen Kernel-Images über dem Image-Builder sind folgende Schritte notwendig:

```
root:/# wget http://downloads.openwrt.org/backfire/10.03-
rc1/rb532/OpenWrt-ImageBuilder-rb532-for-Linux-i686.tar.bz2
root:/#tar xvj OpenWrt-ImageBuilder-rb532-for-Linux-i686.tar.bz2
root:/#/home/OpenWrt-ImageBuilder-rb532-for-Linux-i686# make image
```

Openwrt-SDK

Sourcen von der Backfire-Seite herunter laden und entpacken.

```
root:/# wget http://backfire.openwrt.org/10.03/backfire_10.03
_source.tar.bz2
root:/#tar xvj backfire_10.03_source.tar.bz2
```

oder mit

```
root:/# svn co svn://svn.openwrt.org/openwrt/branches/backfire
```

die aktuellsten Sourcen herunter laden.

Die Openwrt-SDK enthält alle Sourcen um ein eigenes Kernel-Image zu erstellen.

```
root:/#/home/OpenWrt-SDK-rb532-for-Linux-i686# make menuconfig
```

Über die Option "make menuconfig" erhält der Nutzer einen Editor, indem eine Vielzahl an Optionen zum „Backen“ eines neuen Kernels für das jeweilige Board bereitgestellt wird. Nach vollständiger, korrekter Auswahl wird eine Kernel-Config-Datei abgespeichert und über die Eingabe „make“ das Kernel-Image kompiliert.

Openwrt-Toolchain

Ordner der Toolchain in der Kamikaze Openwrt-SDK:

```
root:/# /home/OpenWrt-SDK-rb532-for-Linux-i686/staging_dir/toolchain
-mipsel_gcc-4.3.3_uClibc-0.9.30
```

Nach dem Export der Crosscompiler-Daten kann über die Kommandozeile die Cross-Compile-Funktion mit „mipsel-openwrt-linux-gcc“ überprüft werden.

1.2 Implementierung Openwrt

Serielle Schnittstelle auf RB532

Das RB532 wird über die serielle Schnittstelle mit dem EWR verbunden.

Einstellungen für Minicom (EWR):

```
port          /dev/ttyS0
backspace     BS
baudrate      115200
bits          8
parity        N
stopbits      1
scriptprog    /usr/bin/runscript
```

Löschen von RouterOS

Es erfolgt der Einsteig in den RoutBOOT-2.12 Bootloader am RB532 und die Wahl der Option # e - format nand welche mit „yes“ bestätigt wird. Nun sollte das Board mit „x“ neu gestartet und danach wieder in das Menü

```
RouterBOOT-2.12
What do you want to configure?
d - boot delay
k - boot key
s - serial console
o - boot device
u - cpu mode
f - try cpu frequency
c - keep cpu frequency
r - reset configuration
e - format nand
g - upgrade firmware
i - board info
p - boot protocol
t - do memory testing
x - exit setup
your choice:
```

des RB532-Bootloader eingestiegen werden.

1.3 Zusätzliche Pakete

Paket-Manager am RB532

Der Opkg-Paketmanager bietet die Möglichkeit über ein IP-Netz verschiedene Pakete aus dem Netz nachzuladen.

Ssh und sftp-Paket für den Fernzugriff auf die RB532 Boards.

```
root@OpenWrt:/# opkg install --force-depends openssh-sftp-client
root@OpenWrt:/# opkg install --force-depends openssh-sftp-server
root@OpenWrt:/# opkg install --force-depends openssh-server
root@OpenWrt:/# opkg install --force-depends openssh-client
```

Hinzu kommt das Digitemp-Paket und ein Paket für die Überprüfung der seriellen Schnittstelle.

```
root@OpenWrt:/# opkg install --force-depends digitemp
root@OpenWrt:/# opkg install --force-depends stty
```

Temperaturmessung am RB532

Über die Kommandozeileingabe wird die RS-232 Schnittstelle am RB532 mit dem Paket "stty" überprüft und auf die notwendigen Einstellungen konfiguriert.

```
root@OpenWrt:/# stty -F /dev/ttyS0 9600.
```

Der Zugriff auf den Temperatursensor erfolgt über:

```
# digitemp_DS9097U <Funktionsaufruf>
```

Neukompilierung des Digitemp-Pakets mittels Openwrt-SDK:

```
root:/# wget http://www.digitemp.com/software/linux/
digitemp-3.5.0.tar.gz
root:/# tar zxvf digitemp-3.5.0.tar.gz
root:/# cd digitemp-3.5.0
root:/# nano Makefile
```

Cross-Compilier-Konfiguration:

```
VERSION = 3.5.0
CC       = mipsel-openwrt-linux-gcc
CFLAGS  = -I./src -I./userial -O2 -Wall # -g
```

Neues Paket erstellen:

```
root:/# make ds9097u
```

Das neu erstellte Paket wird auf das RB532 kopiert und kann von dort über

```
root@OpenWrt:/# ./digitemp_DS9097U -i -a -s /dev/ttyS0
```

aufgerufen werden.

Wird ein Temperatursensor an der seriellen Schnittstelle erkannt, erhält man folgende Ausgabe inklusive Temperatur und Zeitstempel.

```
1060E4010208004B : DS1820/DS18S20/DS1920 Temperature Sensor
ROM #0 : 1060E4010208004B
Wrote .digitemprc
Jan 01 02:04:32 Sensor 0 C: 21.38 F: 70.4
```