

# **Package wrap**

## **Version 3.1.3**

Das fli4l-Team

1. September 2007

# Inhaltsverzeichnis

<b>1</b>	<b>Dokumentation des Paketes wrap</b>	<b>3</b>
1.1	WRAP - Support für WRAP System . . . . .	3
1.1.1	Beispielkonfiguration für RRDTOOL . . . . .	4
1.1.2	Hinweise zur WRAP-Benutzung . . . . .	4
<b>A</b>	<b>Anhang zum Paket wrap</b>	<b>5</b>

# 1 Dokumentation des Paketes wrap

## 1.1 WRAP - Support für WRAP System

**OPT\_WRAP** Standard-Einstellung: OPT\_WRAP='no'

Aktiviert die Unterstützung für das WRAP System.

**WRAP\_WATCHDOG** Startet einen Watchdog auf dem WRAP System direkt nach dem Booten. Sollte das System für mehr als 15 Sekunden blockieren wird ein Reset ausgelöst und das WRAP System startet automatisch neu.

**OPT\_WRAPD** Ein Hintergrundprozess prüft, ob der Taster beim WRAP gedrückt wurde und steuert die drei LEDs.

**WRAPD\_KEY\_ACTION** Eine der folgenden Aktionen kann auf Tastendruck ausgelöst werden:

wlan\_toggle = Hoch- bzw. Runterfahren des WLAN.

reset = Reset des Routers

online\_toggle = Umschalten zwischen Online und Offline

" = leer - der Hintergrundprozess wird gestartet und nur die LEDs werden geschaltet.

**WRAPD\_KEY\_SCRIPT** Hier kann das Script angegeben werden, das sich um die Behandlung der Ereignisse beim Tastendruck kümmert. Das dort angegebene Script wird automatisch auf den Router übertragen.

**WRAP\_LED\_x** Den drei vorhandenen LEDs kann zugewiesen werden, welchen Status sie anzeigen sollen. Implementiert sind: 'online', 'wlan', 'ready' und 'user'. Wenn 'user' verwendet wird, ist unbedingt mit der Variablen WRAPD\_USER\_SCRIPT ein Script anzugeben, welches die LED steuern soll. Wird 'wlan' verwendet ist das WLAN-Device in WRAPD\_WLAN\_DEVICE einzutragen.

**WRAPD\_WLAN\_DEVICE** Hier wird das zu überwachende WLAN.Device eingetragen.

**WRAPD\_USER\_SCRIPT** Hier wird das Script angegeben, welches die LED steuern soll. Gesteuert wird eine LED ganz einfach mit:

```
echo LEDx [on|off|blink] >> /var/run/wrap.fifo
```

### 1.1.1 Beispielkonfiguration für RRDTOOL

Ab Kernel 2.6 kann über OPT\_HTTPD die Temperatur angezeigt werden. Ausserdem kann der Verlauf der Temperatur mittels OPT\_RRDTOOL aufgezeichnet werden.

Beispielkonfig für rrdtool:

```
RRDTOOL_x_SOURCE='wrap'  
RRDTOOL_x_COMMENT='WRAP-Daten'  
RRDTOOL_x_OPTIONS='temperatur'  
RRDTOOL_x_VALUE_INTERVAL='1'  
RRDTOOL_x_GRAPH_INTERVAL='5'  
RRDTOOL_x_GRAPH_PATH='/boot/data/wrap'
```

### 1.1.2 Hinweise zur WRAP-Benutzung

#### Serielle Konsole

Bei Verwendung der seriellen Konsole, also des RS232-Anschlusses des WRAP, ist darauf zu achten, dass überall die gleichen Settings verwendet werden. Das WRAP wird mit einer Übertragungsrate von 38400 und den Übertragungseinstellungen 8N1 ausgeliefert. Wenn man das nicht ändern möchte/kann, dann ist in base.txt der Schlüssel SER\_CONSOLE\_RATE entsprechend anzupassen:

```
SER_CONSOLE_RATE='38400'
```

#### USB

Wenn das WRAP um USB-Anschlüsse erweitert werden soll, so ist im USB-Paket der Lowlevel-Treiber 'ohci' auszuwählen. Es muss also

```
USB_LOWLEVEL='ohci'
```

eingestellt werden. Ausserdem muss im BIOS die USB-Unterstützung aktiviert werden. Dazu wird eine serielle Konsole benötigt. Wenn das WRAP bootet, erfolgt zunächst der Memory-Check. Während dieser durchgeführt wird, kommt man durch die Eingabe von 'S' ins BIOS-Setup. Hier werden die aktiven Einstellungen durch zwei '\*' eingerahmt, während die nicht aktiven Einstellungen eingeklammert sind. Somit muss also durch einen Druck auf 'u' USB aktiviert werden. Danach durch 'q' das Setup verlassen und USB ist aktiviert. Weiteres dazu findet sich in der USB-Doku.

## **A Anhang zum Paket wrap**

# Index

OPT\_WRAP, 3

OPT\_WRAPD, 3

WRAP\_LED\_x, 3

WRAP\_WATCHDOG, 3

WRAPD\_KEY\_ACTION, 3

WRAPD\_KEY\_SCRIPT, 3

WRAPD\_USER\_SCRIPT, 3

WRAPD\_WLAN\_DEVICE, 3