

Package embedded Version 3.3.0

Das fli4l-Team
email: team@fli4l.de

27. November 2008

Inhaltsverzeichnis

1. Dokumentation des Paketes embedded	3
1.1. EMBEDDED - Support für Embedded Systeme	3
1.1.1. Beispielkonfiguration für RRDTOOL	4
1.1.2. Hinweise zur Benutzung der verschiedenen eingebetteten Systeme	4
A. Anhang zum Paket embedded	6

1. Dokumentation des Paketes embedded

1.1. EMBEDDED - Support für Embedded Systeme

OPT_EMBEDDED Standard-Einstellung: `OPT_EMBEDDED='no'`

Aktiviert die Unterstützung für das embedded System.

EMBEDDED_TYPE `EMBEDDED_TYPE` beschreibt den Typ des eingebetteten Systems. Momentan werden die Systeme Wrap, Alix, Surfstation und Soekris unterstützt. Teilaspekte wie Leds werden auch auf einer PC-Plattform unterstützt. Gültige Werte für diese Variable sind daher `wrap`, `alix`, `soekris`, `surfstation` und `pc`.

EMBEDDED_WATCHDOG Startet einen Watchdog auf dem EMBEDDED System direkt nach dem Booten. Sollte das System für mehr als 15 Sekunden blockieren, wird ein Reset ausgelöst und das EMBEDDED System startet automatisch neu.

EMBEDDED_DAEMON Ein Hintergrundprozess prüft, ob der Taster beim eingebetteten System gedrückt wurde und steuert bis zu drei LEDs.

EMBEDDED_BUTTON_ACTION Eine der folgenden Aktionen kann auf Tastendruck ausgelöst werden:

`wlan_toggle` = Aktivieren - bzw. Deaktivieren des WLAN.

`reset` = Reset des Routers

`online_toggle` = Umschalten zwischen Online und Offline

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

EMBEDDED_BUTTON_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.

Standard-Einstellung: `EMBEDDED_BUTTON_SCRIPT='/usr/bin/embedded_keyprog'` Dieses Script kann die Actions `'wlan_toggle'`, `'reset'` und `'online'` behandeln.

EMBEDDED_LED_N **EMBEDDED_LED_x** Die verschiedenen Plattformen haben eine unterschiedliche Anzahl von LEDs, die genutzt werden können, um Dinge zu signalisieren. `EMBEDDED_LED_N` spezifiziert die Anzahl der verfügbaren LEDs.

Den einzelnen 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 `EMBEDDED_USER_SCRIPT` ein Script anzugeben, welches die LED steuern soll. Wird `'wlan'` verwendet ist das WLAN-Device in `EMBEDDED_WLAN_DEVICE` einzutragen.

EMBEDDED_WLAN_DEV Hier wird das zu überwachende `WLAN_Device` eingetragen.

EMBEDDED_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/embedded.fifo
```

EMBEDDED_SOEKRIS EMBEDDED_SOEKRIS ist optional. Mit ihr wird festgelegt, welche Treiber das Soekris zum Ansteuern der Hardware benutzt. Gültige Werte sind 'gpio' und 'i2c'.

'gpio' (default), es werden spezielle Treiber je nach Kernel eingebunden. für 2.4er Kernel von <http://soekris.hejl.de/> bzw. für 2.6er Kernel von http://www.meduna.org/sw_gpio_en.html
'i2c' es werden die normalen Kerneltreiber verwendet (Kernel 2.6).

1.1.1. Beispielkonfiguration für RRDTOOL

Je nach Konfiguration und Hardware ist es möglich, über OPT_HTTPD die Temperatur anzeigen zu lassen. Ausserdem kann der Verlauf der Temperatur mittels OPT_RRDTOOL aufgezeichnet werden. Als Quelle für RRD ist dabei der Typ der eingebetteten Systems anzugeben.

Beispielkonfig für rrdtool:

```
RRDTOOL_x_SOURCE='alix'  
RRDTOOL_x_LABEL='Alix-Temperaturen'  
RRDTOOL_x_OPTIONS_N='1'  
RRDTOOL_x_OPTIONS_1='temperature'  
RRDTOOL_x_OPTIONS_1_LABEL='Temperaturen'
```

```
RRDTOOL_x_SOURCE='wrap'  
RRDTOOL_x_LABEL='WRAP-Temperatur'  
RRDTOOL_x_OPTIONS_N='1'  
RRDTOOL_x_OPTIONS_1='temperature'  
RRDTOOL_x_OPTIONS_1_LABEL='Temperatur'
```

1.1.2. Hinweise zur Benutzung der verschiedenen eingebetteten Systeme

Serielle Konsole bei Wrap und Alix Systemen

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

```
SER_CONSOLE_RATE='38400'
```

Netzwerktreiber bei Wrap Systemen

Als Netzwerkkartentreiber ist bei einem Wrap der Typ 'natsemi' auszuwählen.

Netzwerktreiber bei Alix Systemen

Als Netzwerkkartentreiber ist bei einem Alix der Typ 'via-rhine' auszuwählen.

USB bei Wrap und Alix Systemen

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.

USB bei Alix Systemen

Das Alix unterstützt im Gegensatz zum WRAP USB1.1 und USB 2.0. Es sind somit die beiden Lowlevel-Treiber 'ohci' und 'ehci' auszuwählen. Es muss also

`USB_LOWLEVEL='ohci ehci'` eingestellt werden.

Temperatursensor bei ALIX Systemen

Vom ALIX gibt es verschiedene Modelvarianten. Leider sind nicht alle davon mit einen Temperatursensor ausgestattet. Nur die folgenden Modelle haben den Temperatursensor LM86 bestückt: - alix2c2 bzw. alix2d2 - alix2c3 bzw. alix2d3

A. Anhang zum Paket embedded

Index

EMBEDDED_BUTTON_ACTION, [3](#)
EMBEDDED_BUTTON_SCRIPT, [3](#)
EMBEDDED_DAEMON, [3](#)
EMBEDDED_LED_N, [3](#)
EMBEDDED_LED_x, [3](#)
EMBEDDED_SOEKRIS, [4](#)
EMBEDDED_TYPE, [3](#)
EMBEDDED_USER_SCRIPT, [3](#)
EMBEDDED_WATCHDOG, [3](#)
EMBEDDED_WLAN_DEV, [3](#)

OPT_EMBEDDED, [3](#)