

Raspberry Pi mit Raspbian als WEB-Server

Nach der Installation von Lighttpd (im repository vorhanden) mit "sudo apt-get install lighttpd" vom Terminal aus ist sinnvollerweise eine

Konfiguration des LAN auf eine statische IP-Adresse vorzunehmen. Dazu wird in "/etc/network/interfaces" editiert:

```
nach:  iface lo inet loopback
Zeile:  iface etho inet dhcp    ändern in:    iface etho inet static
                                     address 172.16.1.2
                                     netmask 255.255.255.240
                                     gateway 172.16.1.1
                                     dns-nameserver 172.16.1.1
```

Ab Raspbian Streich besser "/etc/network/interfaces" unverändert lassen und die statische IP-Adresse in der „/etc/dhcpd.conf festlegen (nicht in beiden!). Dazu am Ende einfügen und „sudo service dhcpd restart“:

```
interface eth0
static ip_address=172.16.1.2/24
static routers= 172.16.1.1
static domain_name_servers= 172.16.1.1
```

Kontrolle im Terminal mit: "ifconfig -a".

Um Ressourcen zu sparen: Im Menü „Preferences > Raspberry Pi Configuration > System > Boot > To CLI auswählen“ (oder auch mit „sudo raspi-config“), um einen Start von X-Window abzuschalten.

Vorbereitung der Webseiten

In das Verzeichnis „/var/www“ (bei der Installation automatisch erzeugt) die Webseiten kopieren. Ohne weitere Zusätze beherrscht „lighttpd“ statische Webseiten (mit Javascript und css), erstellt aber nicht selbstverständlich einen Index bei einem Link auf ein Verzeichnis ohne „index.htm“ - Datei.

Änderungen in "/etc/lighttpd/ lighttpd.config" mit Editor (es sind nur die hier notwendigen Zeilen angeben):

```
server.modules = (
...
    "mod_accesslog",                #hinzufügen, wenn Accesslog gewünscht
)
...
accesslog.filename      = "/var/log/lighttpd/access.log" #hinzufügen, wenn Accesslog gewünscht
...
server.port              = 80
index-file.names        = ( "index.htm", "index.html" )
```

Danach Restart mit:

```
/etc/init.d/lighttpd restart
```

Soll auf den **Server von außerhalb** zugegriffen werden (nachdem er in unserem internen Netzwerk bereits läuft und leider nur eine dynamische IP vom Provider zugewiesen wurde), muss ein Dienst für dynamisches DNS genutzt werden, um eine feste URL zu erhalten.

Der Router ist dazu so zu konfigurieren, dass Anfragen von außen an den richtigen Host weitergeleitet werden und seine Firewall den entsprechenden Port (Port 80) freigibt.

Diese einfache Installation kann nur statische Seiten veröffentlichen. Lediglich mit JavaScript kann Dynamik realisiert werden. Auch wird für ein Verzeichnis ohne index.htm kein Index mit einer Zusammenstellung der Dateien im Verzeichnis erstellt. So gibt es keine serverseitigen Skripte, die Hacker erreichen könnten. (Bei mir gehen so einige 100 Versuche pro Tag ins Leere, d.h. 404.)

(Im Netz gibt es auch Anleitungen für eine Zusammenarbeit mit PHP und MySQL.)