

Neues Schuhwerk

Der Raspberry Pi 4 lässt sich komplett ohne SD-Karte per Netzwerk booten – nicht nur für eine Host-Target-Entwicklung ein sehr nützliches Feature. Eva-Katharina Kunst, Jürgen Quade

Die Autoren

Eva-Katharina Kunst ist seit den Anfängen von Linux Fan von Open Source. Jürgen Quade, Professor an der Hochschule Niederrhein, führt auch für Unternehmen Schulungen zu den Themen Treiberprogrammierung und Embedded Linux durch.

Baut man die Software für ein anderes System entspannt auf einem leistungsfähigen PC, spricht man von einer Host-Target-Entwicklung. Eine solche vereinfacht sich noch einmal deutlich, wenn die Zielplattform das Booten via Netzwerk unterstützt: Das erspart sowohl das lästige Ablegen der frisch generierten Software auf ein Flash-Speichermedium der Zielplattform als auch den Ärger, der sich ergibt, wenn der Bootvorgang durch einen Fehler missglückt. Dann muss man das verstaubte Speichermedium aufräumen und frisch mit bootfähiger Software bespielen. Das erfordert typischerweise mehr als nur ein oder zwei leichtfüßige Kommandos auf der Konsole. Beim Netzwerkboot dagegen lässt sich das funktionsuntüchtige Kompilat ratzfatz gegen

ein funktionstüchtiges austauschen, und nach dem nächsten Neustart lässt sich die Zielplattform wieder ansprechen.

Auch der Raspberry Pi bietet das Booten per Netzwerk an, man muss ihn dafür nur geeignet präparieren. Das neueste Modell des Mini-Rechners benötigt dazu nicht einmal eine SD-Karte: Die Raspberry Pi Foundation hat den RasPi 4 mit einem EEPROM (Electrical Erasable Programmable Read Only Memory) ausgestattet, das die benötigte Konfiguration aufnimmt. Der Inhalt bleibt auch nach Verlust der Stromversorgung erhalten.

Im Netzwerk selbst, in dem der RasPi per Ethernet-Kabel hängt, stellt ein sogenannter TFTP-Server Abbilddateien von Kernel und Userland (Root-Filesystem) zur Verfügung. Das Userland, also die

Sammlung der Anwendungsprogramme, Bibliotheken und sonstigen benötigten Komponenten, kann man auch per NFS (Network File System) in das System einbinden. Dann steht für alle Bedarfsfälle auch ausreichend Speicher zur Verfügung. Im Bereich eingebetteter Systeme ist es aber üblich und sinnvoll, das Userland im Hauptspeicher abzulegen und sich damit beim Booten eine Prüfung des Dateisystems zu sparen und mögliche Flash-Speicher nicht kaputtzuschreiben. Abgesehen davon kann man so nach dem Start die Netzwerkverbindung kappen, und das System lebt und arbeitet dennoch weiter. Mit einem Verlust der Stromversorgung geht dann freilich auch ein Verlust eventuell veränderter Daten einher. Als naheliegende Gegenmaßnahme speichert man bei Veränderungen daher Daten auf einem dazu erreichbaren Speichermedium (Flash, USB-Stick) ab.

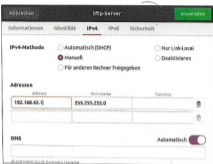
Für das Booten via Netzwerk gilt es, auf der einen Seite den Client geeignet zu konfigurieren, also den Raspberry Pi 4. Auf der anderen Seite benötigt man auf dem Host-System einen TFTP-Server, den es zudem mit den erforderlichen Bootdateien zu betanken gilt.

Let's serve

Unter Ubuntu bietet sich für den TFTP-Server das einfach zu installierende und zu konfigurierende Paket `tftpd-hpa` an. Listing 1 zeigt die dazu notwendigen Kommandos. Der Befehl aus der ersten Zeile sorgt für das Einrichten des Servers sowie des später bei der Fehlersuche nützlichen TFTP-Clients.

Zum Bestücken des TFTP-Servers mit den Bootdateien für den RasPi holen Sie

aus dem zugehörigen Github-Repository die Datei `master.zip` (Zeile 3), die Sie etwa im Verzeichnis `/tmp/` auspacken und dann nach `/srv/tftp/` verschieben (Zeile 5). Ein abschließendes `chmod` (Zeile 6) räumt dem TFTP-Server definitiv Zugriff auf die abgelegten Daten ein. Die müssen Sie noch leicht modifizieren und ergänzen, damit



1 Die Konfiguration der Netzwerkparameter auf dem Host-Rechner gestaltet sich unaufwendig.

Listing 1: TFTP-Server einrichten

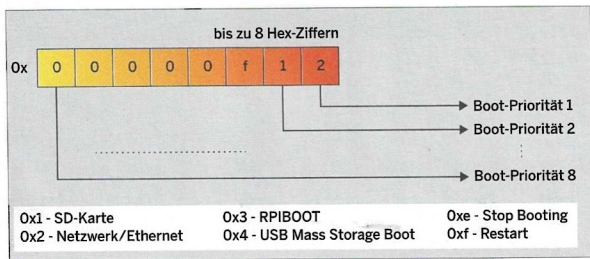
```
01 $ sudo apt -y install tftpd-hpa tftp-hpa
02 $ cd /tmp/
03 $ wget https://github.com/raspberrypi/firmware/archive/master.zip
04 $ unzip master.zip
05 $ sudo mv firmware-master/boot/* /srv/tftp/
06 $ sudo chmod -R 755 /srv/tftp/
```

Listing 2: Server überwachen

```
$ sudo systemctl restart tftpd-hpa
$ tail -f /var/log/syslog | grep tftp
```

Listing 3: Ergänzungen in `config.txt`

```
[all]
enable_uart=1
dtoverlay=disable-bt
kernel=kernel7l.img
initramfs rootfs.cpio
followkernel
```



2 Die Konfiguration von Bootprioritäten erfolgt mithilfe des Parameters BOOT_ORDER.

nicht nur der Kernel geladen wird, sondern auch das Root-Filesystem.

Die Konfiguration des TFTP-Servers findet sich im Verzeichnis /etc, Sie müssen sie aber üblicherweise nicht anfassen. Fürs Debugging kann es sinnvoll sein, dem TFTP-Server in der Datei /etc/default/tftpd-hpa beim Start die Optionen --secure und --verbose mitzugeben. So können Sie nach einem Neustart des Servers (Listing 2, erste Zeile) dessen Aktivitäten beobachten (zweite Zeile).

Verändere dich

Die Datei /srvr/tftp/config.txt enthält Konfigurationsparameter zur Steuerung des Bootvorgangs, die der Bootloader auswertet. Dazu gehören unter anderem Device-Tree-Overlays, über die der Kernel Informationen zur eingesetzten Hardware erhält.

Außerdem lassen sich dort die Namen der zu ladenden Abbilddateien spezifizieren. In Listing 3 sind das die beiden letzten Zeilen, die als Kernel die Datei kernel7l.img (32-Bit-Kernel) und als Userland die Datei rootfs.cpio festlegen. Die beiden (optionalen) Zeilen darüber aktivieren die serielle Schnittstelle, was den im Bereich eingebetteter Systeme immer noch üblichen Zugriff per Terminalemu-

lation ermöglicht. Ergänzen Sie also die vier Zeilen in der Datei config.txt.

Die Datei rootfs.cpio befindet sich übrigens noch nicht in unserem Verzeichnis /srvr/tftp/. Zum Testen kopieren Sie eine von uns per Buildroot gebaute Version aus dem Download-Bereich zu diesem Artikel ins Verzeichnis. Von dort beziehen Sie auch die Abbilddatei kernel7l.img und befördern sie an dieselbe Stelle – dazu später noch mehr.

Früchtekuchen

Um den RasPi geeignet zu konfigurieren, müssen Sie das Netzwerk-Setup kennen. Sie benötigen lediglich die IP-Adresse des TFTP-Servers. Im Detail gilt es, zwei Varianten zu unterscheiden. Im ersten Fall befinden sich sowohl das Host-System mit dem TFTP-Server im lokalen Netz als auch das Zielsystem, also der via Ethernet angebundene Raspberry Pi. Im zweiten Fall hängt der Mini-Rechner ohne den Umweg über ein Gateway direkt per Ethernet am Host-System. Der Unterschied: Im ers-

EEPROM-Konfigurationsparameter (Auswahl)

Parameter	Vorgabe	Bedeutung
BOOT_UART	0	Aktiviert Debug-Ausgaben über die serielle Schnittstelle.
WAKE_ON_GPIO	1	Ermöglicht das Aufwecken aus dem Low-Power-Mode.
POWER_OFF_ON_HALT	0	Versetzt den Raspberry Pi bei einem Halt in den Low-Power-Mode.
BOOT_ORDER	-	Legt die Bootpriorität fest.
MAX_RESTARTS	-1	Anzahl der Restarts, bevor ein Full-Reset erfolgt.
SD_BOOT_MAX_RETRIES	0	Anzahl der Bootversuche über die SD-Karte.
NET_BOOT_MAX_RETRIES	0	Anzahl der Bootversuche via Ethernet.
DHCP_TIMEOUT	-	Timeout in Millisekunden.
DHCP_REQ_TIMEOUT	-	Timeout in Millisekunden.
TFTP_FILE_TIMEOUT	-	Timeout für den Datei-Download (zwischen 5000 und 30 000 Millisekunden).
TFTP_IP	-	IP-Adresse des TFTP-Servers.
TFTP_PREFIX	-	Aktiviert einen gerätespezifischen Datei-Download.
TFTP_PREFIX_STR	-	Verzeichnisname für den gerätespezifischen Datei-Download.
CLIENT_IP	..	IP-Adresse des Raspberry Pi.
SUBNET	..	Subnetzmaske.
GATEWAY	..	IP-Adresse eines Gateways.
MAC_ADDRESS	..	Überschreibt die hinterlegte MAC-Adresse
DISABLE_HDMI	0	Deaktiviert Bootnachrichten per HDMI.

Listing 4: EEPROM-Konfiguration

```
[all]
BOOT_UART=1
WAKE_ON_GPIO=1
POWER_OFF_ON_HALT=1
DHCP_TIMEOUT=30000
DHCP_REQ_TIMEOUT=1000
TFTP_FILE_TIMEOUT=20000
TFTP_IP=192.168.178.223
BOOT_ORDER=0xf12
SD_BOOT_MAX_RETRIES=2
NET_BOOT_MAX_RETRIES=2
CLIENT_IP=192.168.178.203
SUBNET=255.255.255.0
```

ten Fall integrieren Sie den RasPi sorgsam ins lokale Netz, im zweiten Fall bestimmen Sie die Netzwerkparameter selbst.

So könnten Sie sich, um beim zweiten Fall zu bleiben, beispielsweise für die private Netzadresse 192.168.42.0/24 entscheiden und dem Host die IP-Adresse 192.168.42.1 zuweisen. Der RasPi bekommt die IP-Adresse 192.168.42.42. Auf dem Ubuntu-Host wählen Sie dazu die Netzwerkinstellungen aus. Unter dem Feld kabelgebunden erstellen Sie durch Anklicken des kleinen, unscheinbaren Pluszeichens ein neues Profil, in das Sie im Reiter *Identität* einen Namen eingeben (im Beispiel *tftp-server*). Im Reiter *IPv4* wählen Sie die Methode *manuell* aus und geben die Adresse 192.168.42.1 mit der Netzmaske 255.255.255.0 ein. Sobald das Profil aktiviert ist, verifizieren Sie in einer Konsole durch Eingabe von `ip -br a` die Adresszuweisung.

Dasselbe Kommando nutzen Sie auch, um die IP-Adresse herauszufinden, wenn Sie sich per Ethernet mit dem lokalen Netz verbunden haben (Fall eins). Am besten konfigurieren Sie den Router so, dass Ihr Rechner diese IP-Adresse grundsätzlich erhält, sonst bleibt die Freude über das Booten via Netzwerk nur von kurzer Dauer. Im Router legen Sie auch die IP-Adresse für den Raspberry Pi fest, wozu Sie aber klassischerweise die MAC-Adresse des Mini-Rechners benötigen.

Da Sie den RasPi ohnehin auf konventionelle Weise per SD-Karte zur Programmierung des EEPROM booten müssen, nutzen Sie die Gelegenheit, die MAC-Adresse und die standardmäßig vom Router (zum Beispiel einer Fritzbox) zugewiesene Adresse zu evaluieren und im Router zu fixieren. Die Adresse des Kleinstrechners müssen Sie in diesem Szenario übrigens nicht zwangsweise konfigurieren: Sie lässt sich auch beim Booten vom DHCP-Server (Router) zuweisen. Die Vergabe einer festen Adresse erleichtert aber später eine potenziell notwendige Fehlersuche.

Ob nun Fall eins oder zwei vorliegt: Letztlich benötigen Sie zur Konfiguration die IP-Adresse des Host-Systems.

RasPi konfigurieren

Booten Sie den Raspberry Pi 4 mit einem Standard-Pi-OS. Falls noch nicht eingerichtet, finden Sie beispielsweise in der

```
Dec 30 14:41:21 rasp-cscka 16.0f9d 933537: 892 From 192.168.16.282: Filename: config.txt
Dec 30 14:41:21 rasp-cscka 16.0f9d 933538: 892 From 192.168.16.282: Filename: config.txt
Dec 30 14:41:22 rasp-cscka 16.0f9d 933539: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:22 rasp-cscka 16.0f9d 933540: 892 From 192.168.16.282: Filename: rootfs.cpio
Dec 30 14:41:22 rasp-cscka 16.0f9d 933541: 892 From 192.168.16.282: Filename: rootfs.zpio
Dec 30 14:41:81 rasp-cscka 16.0f9d 933542: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933543: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933544: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933545: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933546: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933547: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933548: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933549: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933550: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933551: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933552: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933553: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933554: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933555: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933556: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933557: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933558: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933559: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933560: 892 From 192.168.16.282: Filename: bootfd.txt
Dec 30 14:41:81 rasp-cscka 16.0f9d 933561: 892 From 192.168.16.282: Filename: bootfd.txt
```

3 Das Syslog verschafft einen Überblick über den Bootvorgang.

RasPi-Dokumentation  eine Anleitung zur Installation. Loggen Sie sich nach dem Booten ein.

Eventuell müssen Sie noch das Programm `Rpi-eeprom` per `apt install rpi-eeprom` installieren. Mit ihm lassen

sich die Konfigurationsparameter durch den Aufruf `sudo -E rpi-eeprom-config --edit` programmieren. Geben Sie die Parameter wie in Listing 4 gezeigt ein. Dabei passen Sie die IP-Adressen an, wobei Sie die Zeile mit der `CLIENT_IP` in den meis-

```
5.512952] bcmgenet fe580000.ethernet eth0: Link is Up - 100Mbps/Full - Flow control rx/tx
vdhcpd: sending discover
vdhcpd: lease select for 192.168.16.280
vdhcpd: lease of 192.168.16.200 obtained, lease time 86400
Selecting router:
  0 8.030699] random: mktemp: uninitialized urandom read (6 bytes read)
adding dns 192.168.16.1
OK
Starting ntpd: OK
Starting dropbear sshd: OK
Starting mosquitto: OK
  8.374614] brcmfmac: brmf_cfg80211_set_power_mgmt: power save enabled
successfully initialized wpa_supplicant
  8.595103] random: wpa_supplicant: uninitialized urandom read (4 bytes read)
  8.603311] random: wpa_supplicant: uninitialized urandom read (4 bytes read)
Starting dnsmasq: OK
Firewall activated...
nftt-publishing activated...

Linux-Magazin welcomes You
Kiste login: █
```

4 Nach einem Neustart klappt nun das Login auf dem Testsystem.

```
Linux-Magazin welcomes You
Kiste login: root
Pass: █
-----
A - Serieller Anschluss      : /dev/ttyUSB0
B - Pfad zur Lockdatei      : /var/lock
C - Programm zur Rufannahme:
D - Programm zum Wählen    :
E - Bsp/Par/Ports           : 115200 8N1
F - Hardware Flow Control   : Nein
G - Software Flow Control   : Nein
-----
Welchen Parameter möchten Sie ändern? █
  Bildschirm und Tastatur
  Speichern als wFl+
  Einstellungen speichern als ...
  Verlassen
```

Ctrl-A ? for help | 115200 8N1 | MOR | MiniCom 2.7.1 | VT102 | OFFLine | ttyUSB0

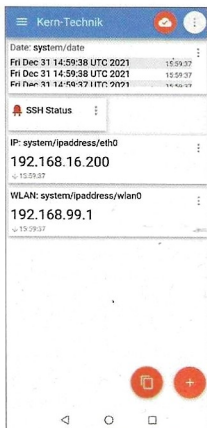
5 Die Minicom-Konfiguration für den Zugriff per serieller Schnittstelle.

ten Setups wie beschrieben auch weglassen können. Nach dem Abspeichern der Konfiguration und dem Verlassen des Programms steht zwingend ein Neustart über den Befehl `reboot` auf der Konsole an. Erst damit landet die Konfiguration tatsächlich im EEPROM. In Listing 4 finden sich auch einige Timing-Parameter. Möglicherweise können Sie in Ihrem Setup darauf verzichten, wir haben damit aber in unterschiedlichen Umgebungen ein zuverlässigeres Bootverhalten erzielt.

Über die Parameter `BOOT_ORDER`, `SD_BOOT_MAX_RETRIES` und `NET_BOOT_MAX_RETRIES` optimieren Sie gegebenenfalls das Bootverhalten. Mit `BOOT_ORDER` konfigurieren Sie die Bootpriorisierung. Der abzuspichernde Wert umfasst bis zu acht Hex-Ziffern, die der RasPi von rechts nach links interpretiert [2]. Die in Listing 4 verwendete Kombination `0xf12` besagt, dass der RasPi zunächst versucht, per Netzwerk zu booten (`0x2`). Schlägt das fehl, greift Priorität 2, also das Booten per SD-Karte (`0x1`). Sollte auch dieser Versuch nicht von Erfolg gekrönt sein, fängt der RasPi wieder bei Priorität eins an (`0xf`). Der Code `0xf12` priorisiert also beim Booten das Netzwerk, selbst wenn eine SD-Karte im Slot steckt. Eine Beschreibung der wichtigsten Parameter finden Sie in der Tabelle EEPROM-Konfigurationsparameter (Auswahl), weitere Informationen liefert die RasPi-Dokumentation [3].

Stunde der Wahrheit

Um das Booten via Netzwerk zu testen, fahren Sie den Raspberry Pi herunter, kap-



6 Abruf von Informationen via via MQTT über das WLAN.

pen die Stromversorgung und entnehmen die SD-Karte. Anschließend setzen Sie den Mini-Rechner wieder unter Strom. Auf dem Host – also auf dem Rechner, auf dem der TFTP-Server läuft – starten Sie eine Konsole und sehen sich mithilfe des Kommandos `tail -f /var/log/syslog | grep tftp` die Log-Ausgaben an, die in etwa so ausfallen sollten wie in Abbildung 8. Geben Sie dem Bootvorgang etwas Zeit, insbesondere wenn Kernel und Root-Dateisystem (Userland) etwas umfangreicher ausfallen.

Hat der RasPi gebootet, können Sie sich mit ihm verbinden [4]. Username und Passwort lauten `root`. Dazu verwenden Sie die serielle Schnittstelle und eine Terminalemulation wie Minicom. Typischerweise starten Sie Minicom auf dem Ubuntu-Rechner mit der Option `-D /dev/ttyUSB0`. Vergessen Sie nicht, *Hardware Flow Control* auf *Nein* zu setzen [5].

Besser loggen Sie sich statt über die serielle Schnittstelle per SSH auf dem frisch gebooteten RasPi ein. Dazu benötigen Sie dessen IP-Adresse, die Ihnen entweder der Router oder der Netzwerks scanner Nmap verraten. Es gibt einen weiteren Weg, die IP-Adresse zu evaluieren. Dazu laden Sie aus dem Download-Bereich zu diesem Artikel noch die Abbilddatei `kerne171.1img` herunter und legen sie im Verzeichnis `/srvr/tftp/` ab. Mit dem passenden Kernel spannt der Raspberry Pi 4 ein WLAN unter dem Namen `Linux-Magazin` auf. Das Passwort des drahtlosen Netzes lautet `kern-technik`.

Falls Sie sich mit MQTT [3] auskennen, finden Sie nun unter dem Topic `system/ipaddress/eth0` die IP-Adresse im lokalen Netz, unter `system/ipaddress/wlan0` lassen Sie sich die IP-Adresse im WLAN anzeigen. Das Topic `system/date` liefert die aktuelle Systemzeit, sofern der RasPi Netzzugriff hat. Den Screenshot in Abbildung 6 haben wir mit der Android-App `IoTMQTTPanel` erstellt. Die zur Konfigura-

tion des MQTT-Servers benötigte IP-Adresse lautet im Übrigen `192.168.99.1`.

Troubleshooting

Gibt der Raspberry Pi nach dem Neustart kein Lebenszeichen von sich, ist Troubleshooting angesagt. Testen Sie als Erstes den TFTP-Server mithilfe des TFTP-Clients, den Sie eingangs auf dem Host mit eingerichtet haben. Listing 5 zeigt die Kommandos, mit denen Sie die Datei `config.txt` herunterladen. Lassen Sie in einem separaten Terminalfenster den Befehl `tail -f /var/log/syslog | grep tftp` mitlaufen. Funktioniert der TFTP-Server und sehen Sie im Log Ausgaben, setzen Sie den verkabelten RasPi unter Strom.

Falls keine Log-Nachrichten erscheinen, liegt wohl ein Netzwerkproblem vor. Überprüfen Sie dann die auf dem RasPi konfigurierten IP-Adressen. Über einen per HDMI am Mini-Rechner angeschlossenen Monitor erhalten Sie beim Booten eingeschränkt Informationen. Insbesondere können Sie ersehen, ob eine Netzverbindung existiert, und wenn ja, welche IP-Adresse der Raspberry Pi verwendet. Weitergehende Infos bekommen Sie mit einem USB-to-Serial-Adapter. Haben Sie den richtig angeschlossen und die Terminalemulation (Minicom) geeignet konfiguriert, können Sie wesentliche Teile des Bootvorgangs auf dem RasPi verfolgen.

Fazit

Die Konfiguration des TFTP-Servers und das Einrichten des Raspberry Pi zum Netzwerkboot nimmt etwas Zeit in Anspruch. Ist das aber einmal erfolgreich erledigt, bootet der RasPi recht zuverlässig. Gerade bei der Host-Target-Entwicklung will man das Feature schon bald nicht mehr missen. (jlu) ■

Listing 5: config.txt holen

```
$ tftp 192.168.42.1
tftp> get config.txt
tftp> quit
```

Dateien zum Artikel heruntergeladen unter

www.lm-online.de/dl/44781



Weitere Infos und interessante Links

www.lm-online.de/gr/44781