

Ubuntu

debootstrap

Schritte um ein Ubuntu-system anzulegen (armhf als Architektur):

<https://help.ubuntu.com/lts/installation-guide/armhf/apds04.html>

```
sudo apt-get install qemu-user-static debootstrap binfmt-support

targetdir=$(pwd)/debootstrap_ubuntu_18.4
distro=bionic
arch=armhf


sudo debootstrap --arch=$arch --foreign $distro $targetdir

#wenn folgender Fehler kommt: E: Cannot install into target '...' mounted
with noexec or nodev
sudo mount -i -o remount,exec,dev /mounted_dir

sudo cp /usr/bin/qemu-arm-static $targetdir/usr/bin/
sudo cp /etc/resolv.conf $targetdir/etc
sudo distro=$distro chroot $targetdir
export LANG=C
/debootstrap/debootstrap --second-stage

#echo "deb-src http://archive.ubuntu.com/ubuntu $distro
main">>/etc/apt/sources.list
#echo "deb http://security.ubuntu.com/ubuntu $distro-security
main">>/etc/apt/sources.list
#echo "deb-src http://security.ubuntu.com/ubuntu $distro-security
main">>/etc/apt/sources.list
echo "deb http://ports.ubuntu.com/ubuntu-ports/ $distro
main">>/etc/apt/sources.list
echo "deb-src http://ports.ubuntu.com/ubuntu-ports/ $distro
main">>/etc/apt/sources.list
echo "deb http://ports.ubuntu.com/ubuntu-ports/ $distro-updates
main">>/etc/apt/sources.list
echo "deb-src http://ports.ubuntu.com/ubuntu-ports/ $distro-updates
main">>/etc/apt/sources.list
echo "deb http://ports.ubuntu.com/ubuntu-ports/ $distro-security
main">>/etc/apt/sources.list
echo "deb-src http://ports.ubuntu.com/ubuntu-ports/ $distro-security
main">>/etc/apt/sources.list

echo "bpi-r2-ubuntu" >/etc/hostname
#root-password setzen! sonst kein Login möglich
passwd
```

fstab/... konfigurieren wie bei [debian](#)  ubuntu 18.4 verwendet netplan.io als Standard Netzwerk-Framework [bionic releasenotes](#)

```
#chroot verlassen  
exit
```

packen:

```
cd $targetdir  
sudo tar cvpzf ../debootstrap_$distro.tar.gz .
```

System Auf SD-Karte installieren

SD-Karte vorbereiten

```
#in root-partition entpacken with  
sudo tar -xpf /path/to/debootstrap_$distro.tar.gz  
sudo mkdir lib/modules/  
cd lib/modules/  
#kernel-module hier entpacken
```

temporäre Netzwerk-Konfiguration:

```
ip a  
ip link set eth0 up  
ip addr add 192.168.0.11/24 dev lan0 #freie ip/prefix deines Lan-Segments  
ip link set lan0 up  
ip route add default via 192.168.0.10 #ip deines Routers  
echo "nameserver 192.168.0.10" >>/etc/resolv.conf #ip deines Router für DNS-Auflösung
```

Installieren von Paket "ifupdown" und Hinzufügen von "netcfg/do_not_use_netplan=true" zu den bootopts in der /boot/bananapi/bpi-r2/linux/uEnv.txt Nach einem Reboot wird das „alte“ System mit /etc/network/interfaces verwendet. nun wird die /etc/resolv.conf bei jedem Neustart zurückgesetzt

```
root@bpi-r2-ubuntu:~# ls -l /etc/resolv.conf  
lrwxrwxrwx 1 root root 39 Jun 13 10:27 /etc/resolv.conf ->  
../run/systemd/resolve/stub-resolv.conf  
#löschen des Symlinks und ersetzen durch eine "normale" Datei mit den  
Einstellungen wie bei Debian  
rm /etc/resolv.conf  
echo "nameserver 192.168.0.10" >>/etc/resolv.conf
```

unter ubuntu 18.4 läuft bereits ein eigener dns-dienst, welcher deaktiviert werden muss (gefolgt von einem reboot oder via stop beenden), um z.B. DNSMasq laufen zu lassen (wie in meinem wifi.sh-script)

```
systemctl disable systemd-resolved
systemctl stop systemd-resolved
```

image erstellen

```
imgfile=/path/to/ubuntu-18.04-bpi-r2-preview.img
sudo dd if=/dev/sdx of=$imgfile
#Status von DD über anderes Terminal mit "sudo kill -SIGUSR1 $(pidof dd)"
ermitteln
#image beschränken auf das Ende der Letzten Partition...Pfad zur img-Datei
darf keine Leereichen enthalten
IFS=$'\t' #zur Sicherheit (ignoriere Leerzeichen im Pfad)
ENDOFDATA=$(fdisk -l "$imgfile" |tail -1|awk '{print $3}')
echo $ENDOFDATA
truncate --size=$((($ENDOFDATA+1)*512) $imgfile
#check size
ls -lh "$imgfile"

#Image weiter manipulieren
loopdev=$(losetup -f)
sudo losetup $loopdev $imgfile
sudo partprobe $loopdev
sudo mount ${loopdev}p2 /mnt
ls /mnt
#...
#Freien Speicher mit 0 füllen für besseres Packen
sudo sh -c 'cat /dev/zero >/mnt/null.dat'
sudo rm /mnt/null.dat
sudo umount /mnt

#image packen
gzip $imgfile
md5sum $imgfile.gz > $imgfile.gz.md5
```

Ich habe mein Image auf mein [gdrive](#) hochgeladen zum testen (ubuntu-18.04-bpi-r2-preview.img.gz)

SSH

ssh-server ist in meinem image installiert [ubuntu-18.04-bpi-r2-preview.img.gz](#) from my [gdrive](#), aber root-login muss aktiviert werden

```
echo "PermitRootLogin yes" >> /etc/ssh/sshd_config
service sshd restart
```

es sollten neue host-keys für ssh erzeugt werden...

```
#alte Schlüssel löschen
```

```
rm /etc/ssh/ssh_host_*  
#sshserver-paket neu konfigurieren  
dpkg-reconfigure openssh-server
```

From:

<https://www.fw-web.de/dokuwiki/> - **FW-WEB Wiki**

Permanent link:

<https://www.fw-web.de/dokuwiki/doku.php?id=bpi-r2:ubuntu>

Last update: **2023/06/08 17:06**

