

# Zusammenbau Assembly

Hans Buchmann FHNW/ISE

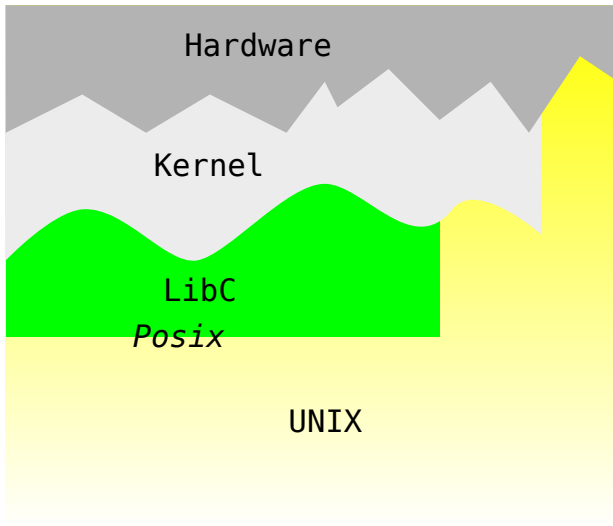
13. November 2018

# Um was geht es ?

Ein erstes vollständiges System

- ▶ Bootloader U-Boot
- ▶ **kernel**
- ▶ UNIX

## Die Schichten



## Das Ziel für BBB

Nach dem Reset:

1. U-Boot startet **kernel**
2. **kernel** startet UNIX
3. UNIX
  - ▶ konfiguriert *ethernet über USB*
  - ▶ startet `ssh` Server
  - ▶ verbindet sich per Wi-Fi

## Was wir schon haben

Toolchain: download

U-Boot : selber gemacht (siehe 4-uboot)

**kernel:** selber gemacht

root Filesystem: download

▶ libc/UNIX

## Die Partitionen und Filesysteme

p1 bootfs:vfat  $\approx 20MiB$

- ▶ U-Boot

- ▶ MLO

- ▶ u-boot.img

p2 rootfs:ext4  $\approx 200MiB$

- ▶ etc/init.d/rcS init-script

- ▶ **kernel**

- ▶ /boot/zImage

- ▶ boot/am335x-boneblack-wireless.dtb

# U-Boot

## Wichtige Befehle

- ▶ `boot startet bootcmd`
- ▶ `ext4load mmc 0:2 addr file`
- ▶ `setenv key value`
- ▶ `run script`
- ▶ `saveenv`

**Remark:** Siehe [www.denx.de/wiki/view/DULG/UBootCmdGroupEnvironment](http://www.denx.de/wiki/view/DULG/UBootCmdGroupEnvironment)

# U-Boot

## Wichtige Variablen

- ▶ bootcmd für U-Boot boot

```
run kernel;bootz 0x82000000 - 0x88000000
```

- ▶ bootargs für den **kernel**

```
root=/dev/mmcblk0p2 rw rootdelay=1 init=linuxrc console=tty00,115200n8
```



# U-Boot

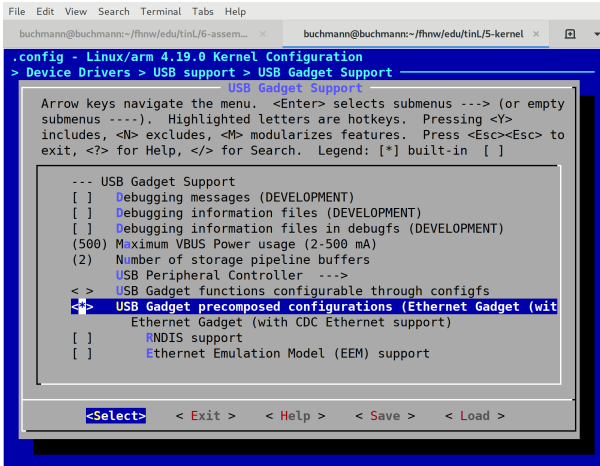
## Wichtige Files

### Auf der Partition p2

- ▶ `/boot/zImage`
- ▶ `/boot/am335x-boneblack-wireless.dtb`

# Konfiguration

## USB-Gadget Support



```
File Edit View Search Terminal Tabs Help
buchmann@buchmann:~/fhnw/edu/tinL/6-assem... x buchmann@buchmann:~/fhnw/edu/tinL/5-kernel x
.config - Linux/arm 4.19.0 Kernel Configuration
> Device Drivers > USB support > USB Gadget Support
    USB Gadget Support
    Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
    submenu ----). Highlighted letters are hotkeys. Pressing <Y>
    includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
    exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

    --- USB Gadget Support
    [ ] Debugging messages (DEVELOPMENT)
    [ ] Debugging information files (DEVELOPMENT)
    [ ] Debugging information files in debugfs (DEVELOPMENT)
    (500) Maximum VBUS Power usage (2-500 mA)
    (2) Number of storage pipeline buffers
    USB Peripheral Controller --->
    < > USB Gadget functions configurable through configs
    <+> USB Gadget precomposed configurations (Ethernet Gadget (wit
        Ethernet Gadget (with CDC Ethernet support)
    [ ] RNDIS support
    [ ] Ethernet Emulation Model (EEM) support

    <Select> < Exit > < Help > < Save > < Load >
```

# Init Script

`target-root-version.tar.gz`

- ▶ `/etc/init.d/rcS` das *Init-Script*
- ▶ `ifconfig` für Internet
- ▶ `sshd` Server für Verbindung

## Workflow Notationen

*sd-card* die Partition vom rootfs auf der SD Karte

*target-root-VERSION.tar.gz* das heruntergeladene rootfs

*target-root* das rootfs von **BBB** auf dem *Host*

## Workflow

### schrittweise Verbesserung

1. Initialer Download `target-root-VERSION.tar.gz`
2. `target-root`
  - ▶ `tar -xf target-root-VERSION.tar.gz -C target-root`
3. Transfer auf `sd-card`
  - ▶ `rsync -av target-root/ sd-card/`
  - ▶ `sync`
4. Test/Konfiguration auf dem **BBB**
5. Update auf dem *Host*
  - ▶ `rsync -av sd-card/ target-root/`
6. → 4

# Die Files

## Partition 1: vfat

- ▶ MLO
- ▶ u-boot.img

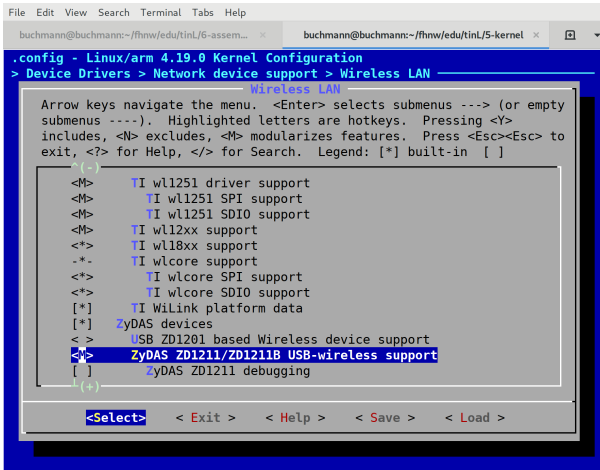
## Partition 2: ext4

- ▶ rootfs auf dem *Host*
- ▶ `rsync -av target-root/ sd-card/`
- ▶ `sync`
- ▶ `/boot/zImage`
- ▶ `/boot/am335x-boneblack-wireless.dtb`

# Ziele Wi-Fi

- ▶ Konfiguration: **kernel**
- ▶ Konfiguration: wi-fi Zugang
- ▶ schrittweises Vorgehen

# Konfiguration



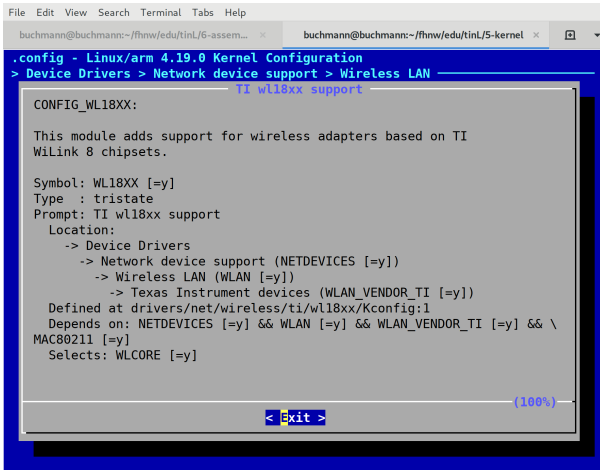
The screenshot shows a terminal window with the Linux Kernel Configuration tool. The title bar indicates the current configuration is for 'Linux/arm 4.19.0 Kernel Configuration'. The navigation path is 'Device Drivers > Network device support > Wireless LAN'. A help text box explains the navigation: 'Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [\*] built-in [ ]'. The menu lists several options, with 'ZyDAS ZD1211/ZD1211B USB-wireless support' highlighted. At the bottom, navigation buttons are visible: '<Select>', '< Exit >', '< Help >', '< Save >', and '< Load >'.

```
File Edit View Search Terminal Tabs Help
buchmann@buchmann:~/fhnw/edu/tinL/6-assem... x buchmann@buchmann:~/fhnw/edu/tinL/5-kernel x
.config - Linux/arm 4.19.0 Kernel Configuration
> Device Drivers > Network device support > Wireless LAN
Wireless LAN
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
^(-)
<M> TI wl1251 driver support
<M> TI wl1251 SPI support
<M> TI wl1251 SDIO support
<M> TI wl12xx support
<?> TI wl18xx support
-*. TI wlcore support
<?> TI wlcore SPI support
<?> TI wlcore SDIO support
[*] TI WiLink platform data
[*] ZyDAS devices
< > USB ZD1201 based Wireless device support
<M> ZyDAS ZD1211/ZD1211B USB-wireless support
[ ] ZyDAS ZD1211 debugging
^(+)
<Select> < Exit > < Help > < Save > < Load >
```

► Test: `dmesg | grep wl`



## Abhängigkeiten



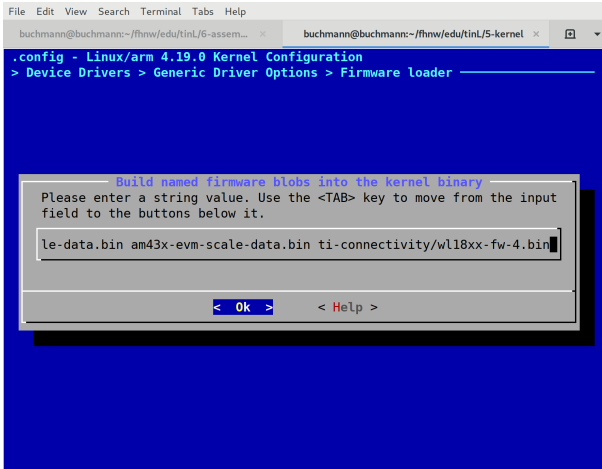
```
File Edit View Search Terminal Tabs Help
buchmann@buchmann:~/fhnw/edu/tinl/6-assem... x buchmann@buchmann:~/fhnw/edu/tinl/5-kernel x
.config - Linux/arm 4.19.0 Kernel Configuration
> Device Drivers > Network device support > Wireless LAN
    TI wl18xx support
CONFIG_WL18XX:

This module adds support for wireless adapters based on TI
WiLink 8 chipsets.

Symbol: WL18XX [=y]
Type : tristate
Prompt: TI wl18xx support
Location:
    -> Device Drivers
        -> Network device support (NETDEVICES [=y])
            -> Wireless LAN (WLAN [=y])
                -> Texas Instrument devices (WLAN_VENDOR_TI [=y])
Defined at drivers/net/wireless/ti/wl18xx/Kconfig:1
Depends on: NETDEVICES [=y] && WLAN [=y] && WLAN_VENDOR_TI [=y] && \
MAC80211 [=y]
Selects: WLCORE [=y]

(100%)
< Exit >
```

## Firmware



```
File Edit View Search Terminal Tabs Help
buchmann@buchmann:~/fhnw/edu/tinl/6-assem... x buchmann@buchmann:~/fhnw/edu/tinl/5-kernel x
.config - Linux/arm 4.19.0 Kernel Configuration
> Device Drivers > Generic Driver Options > Firmware loader

Build named firmware blobs into the kernel binary
Please enter a string value. Use the <TAB> key to move from the input
field to the buttons below it.
le-data.bin am43x-evm-scale-data.bin ti-connectivity/wl18xx-fw-4.bin

< Ok > < Help >
```

# Test

## wlan0

- ▶ `dmesg | grep wl`
- ▶ `ip link set wlan0 up`
- ▶ `iw wlan0 scan`

# WPA

wpa\_supplicant, wpa

- ▶ Konfiguration:
  - ▶ Siehe *3-network*
- ▶ Process:
  - ▶ `wpa_supplicant -D wext -i wlan0 -c path_to_config`
- ▶ Bedienung (funktioniert noch nicht)
  - ▶ `wpa_cli -s wpa_client_socket_file_path`

# DHCP

## manuell

- ▶ `udhcpc -v -i wlan0`
- ▶ `ifconfig wlan0 ip`
  - ▶ `ip` abgelesen von `udhcpc -v -i wlan0`

## automatisch/callback

```
#!/bin/sh
#-----
#on-udhcpc.sh
#(c) H.Buchmann FHNW 2018
#-----
echo "----- on-udhcpc.sh ${1}"
case ${1} in
  defconfig)
    echo defconfig----- ${interface} ${ip};;
    bound)
      ifconfig ${interface} ${ip};;
  #set route here
esac
```

## route/dns

### ▶ route

▶ `route add default gw gw-ip wlan0`

### ▶ DNS

▶ `/etc/resolv.conf:`  
`nameserver 147.86.4.21`  
`#try nameserver 8.8.8.8`

# Aufgabe

## kernel

- ▶ Ethernet über USB
- ▶ Wi-Fi

## UNIX

- ▶ usb0
- ▶ sshd
- ▶ wlan0
- ▶ WPA
- ▶ DHCP
- ▶ DNS

# Aufgabe

Init //etc/init.d/rcS

▶ ergänzen



## Ein paar tools

- ▶ `touch` - change file timestamps
- ▶ `chown` - change file owner and group

## sshd

- ▶ sshd re-exec requires execution with an absolute path
- ▶ Privilege separation user sshd does not exist
- ▶ create group root
  - ▶ `addgroup -g 0 -S root`
- ▶ create user root
  - ▶ `adduser -h /home/root/ -s /bin/sh -G root -S -u 0 root`
- ▶ create group/user sshd
  - ▶ `addgroup sshd`
  - ▶ `adduser -D -H -G sshd sshd`
- ▶ create key
  - ▶ `ssh-keygen -t rsa -f /etc/ssh_host_rsa_key`
- ▶ File `/var/empty` gehört root
- ▶ File `/etc/sshd.config`
  - ▶ `PermitRootLogin yes`