

# Teil I

## Kernel

# Kernel

Hans Buchmann FHNW/ISE

3. Oktober 2019

## Ziele

### Neuer **kernel** auf **BeagleBoneWireless**

- ▶ Download
- ▶ Setup
- ▶ Konfiguration
- ▶ Kompilation
- ▶ Installation

## The Big Picture

grosses Projekt

**Gegeben** Eine grosse Anzahl *source* Files

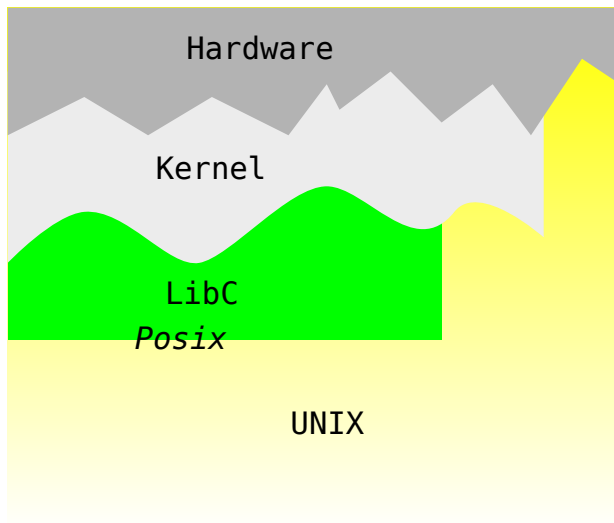
**Gesucht** 2 Files:

- ▶ das **Kernellimage**
- ▶ der *Devicetree*

**Lösung** Klassisches Verfahren

- ▶ Toolchain
- ▶ Makefile

## Die Schichten



## Kernel

### Grosses Projekt

#### Was ist einfach ?

- ▶ **kernel** hängt nicht von anderen Software Komponenten ab
  - ▶ stand alone
- ▶ Braucht nur `make` und *toolchain*

#### Was ist schwierig ?

- ▶ Konfiguration
  - ▶ Wahl der richtigen *source* Files für das Image

[github.com/beagleboard/linux](https://github.com/beagleboard/linux)

Mehrere Möglichkeiten

- ▶ das ganze git repository
- ▶ → nur die letzten  $n$  Versionen `--depth= $n$` 
  - ▶ `git clone -b4.19 --depth=4 path_to_repository`
- ▶ zip File

## Tools

**toolchain** `sourceforge.net/projects/fhnw-tinl/files`

- ▶ `tc-tinl-gcc-8.1.0-2018.05.21.tar.gz`
  - ▶ auf `drive.switch.ch/index.php/s/A6H382zEGDrgfAL`
- ▶ Prefix: `arm-linux-gnueabihf-`
  - ▶ beschreibt:
    - ▶ Architektur: `armv7`
    - ▶ **A**pplication **B**inary Interface: `gnueabihf`

**make** Normales `make`

- ▶ **kernel** Herstellung:
  - ▶ `make cmd`



## Wo ist was ?

```
tinL
├── 5-kernel
│   ├── build ..... generated kernel files
│   │   ├── .config ..... die aktuelle Konfiguration
│   ├── tools ..... for making
│   │   ├── kernel.sh ..... wrapper to kernel Makefile
│   ├── config
│   │   ├── config.sh ..... for kernel.sh
│   │   ├── kernel.config ..... 'gute' kernel Konfiguration
│   ├── tc ..... link to toolchain
├── resources
│   ├── beaglebone-black
│   │   ├── linux ..... the source tree
```

## Erste Konfiguration

- ▶ Hilfe
  - ▶ `./tools/kernel.sh help`
- ▶ Vordefinierte Konfiguration
  - ▶ `./tools/kernel.sh bb.org_defconfig`
- ▶ Anpassung der Konfiguration
  - ▶ `./tools/kernel.sh menuconfig`
  - ▶ `./tools/kernel.sh xconfig`

# Kompilation

- ▶ `./tools/kernel.sh zImage`
  - ▶ erzeugt `build/arch/arm/boot/zImage`
- ▶ `./tools/kernel.sh dtbs`
  - ▶ erzeugt `build/arch/arm/boot/dts/am335x-boneblack-wireless.dtb`  
*Devicetree*

Remark: *Devicetree* später behandelt

# Installation auf SD-Card

## ► Kopiere

**Image** build/arch/arm/boot/zImage

**Devicetree** build/arch/arm/boot/dts/am335x-boneblack-wireless.dtb  
auf

## ► SD-Card Partition 2:Root File System

```
/ ..... Partition 2
├── boot
│   ├── zImage
│   └── am335x-boneblack-wireless.dtb
```

## Workflow

### schrittweise Herstellung

#### 0 Setup der Toolchain

#### 1 Default Konfiguration (falls vorhanden)

- ▶ `sh tools/kernel.sh bb.org_defconfig`

#### 2 Herstellung

- ▶ `tools/kernel.sh zImage`

#### 3 Transfer/Start/Test auf **BeagleBoneWireless**

- ▶ Partition 2 `/boot/`

#### 4 (Re)Konfiguration

- ▶ `sh tools/kernel.sh menuconfig`

→ 2 `ev. cp build/.config config/kernel.config`

# Workflow

## ▶ **BeagleBoneWireless** *default* Konfiguration

- ▶ herstellen
- ▶ auf SD-Karte
- ▶ ausprobieren

## ▶ Die *default* Konfiguration ändern:

- ▶ Internet über USB:

Device Drivers

└─ USB support

└─ USB Gadget Support

└─ USB Gadget Drivers

- ▶ nur eine CPU
- ▶ keine ALSA Soundkarte
- ▶ ...

## Die beteiligten Files

- ▶ `zImage` der *kernel*
- ▶ `am335x-boneblack-wireless.dtb` der *device tree*

## Teil II

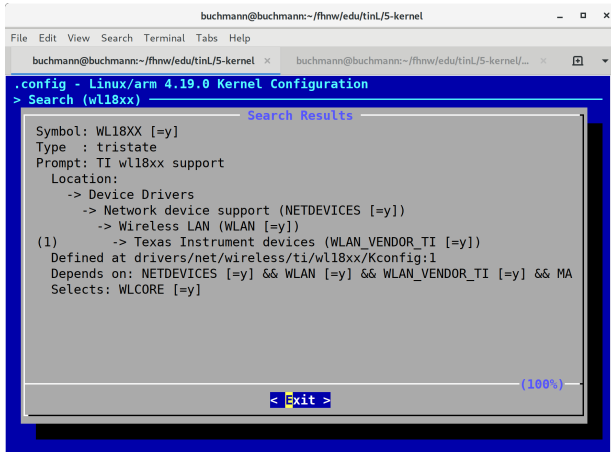
# WiFi-Kernel



## Konfiguration für WiFi

- ▶ Kern konfigurieren
  - ▶ firmware
- ▶ WiFi aufsetzen
  - ▶ Siehe **3-network** Wi-Fi

# Konfiguration Driver

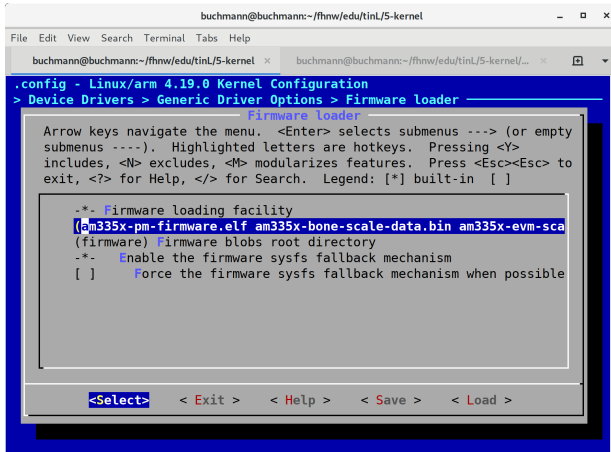


The screenshot shows a terminal window titled "buchmann@buchmann: ~/fhnw/edu/tinL/5-kernel". The terminal is running the "config" utility for "Linux/arm 4.19.0 Kernel Configuration". The user has entered the command "> Search (wl18xx)". The search results are displayed in a box with the following content:

```
Symbol: WL18XX [=y]
Type : tristate
Prompt: TI wl18xx support
Location:
  -> Device Drivers
    -> Network device support (NETDEVICES [=y])
      -> Wireless LAN (WLAN [=y])
        (1) -> 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] && MA
          Selects: WLCORE [=y]
```

At the bottom of the search results box, there is a blue bar with the text "< Exit >" and a progress indicator "(100%)".

# Konfiguration Firmware



The screenshot shows a terminal window titled "buchmann@buchmann: ~/fhnw/edu/tinL/5-kernel". The terminal displays the "Linux/arm 4.19.0 Kernel Configuration" menu. The navigation path is "> Device Drivers > Generic Driver Options > Firmware loader". The "Firmware loader" sub-menu is open, showing instructions and configuration options. The option "am335x-pm-firmware.elf am335x-bone-scale-data.bin am335x-evm-sca" is highlighted. The bottom of the window shows navigation buttons: "< Select >", "< Exit >", "< Help >", "< Save >", and "< Load >".

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

    -* Firmware loading facility
    [am335x-pm-firmware.elf am335x-bone-scale-data.bin am335x-evm-sca
    (firmware) Firmware blobs root directory
    -* Enable the firmware sysfs fallback mechanism
    [ ] Force the firmware sysfs fallback mechanism when possible

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

## Wi-Fi aufsetzen

- ▶ target-root-2018.10.24.tar.gz auf Partition 2 der SD Karte
- ▶ eigener *kernel* auf Partition 2 der SD Karte `/boot`
- ▶ Siehe **3-network**

# Wi-Fi

## Basics

- ▶ **3-network** Seite 14

## Connect

- ▶ **3-network** Seite 15

## Internet

- ▶ **3-network** Seite 16: `udhcpc -i wlan0`
- ▶ **3-network** Seite 12: route, DNS