Teil I

Kernel

Kernel

Hans Buchmann FHNW/ISE

11. März 2020

Ziele

Neuer **kernel** auf **BeagleBoneGreen**

- Download
- Setup
- Konfiguration
- Kompilation
- Installation

The Big Picture

grosses Projekt

Gegeben Eine grosse Anzahl *source* Files

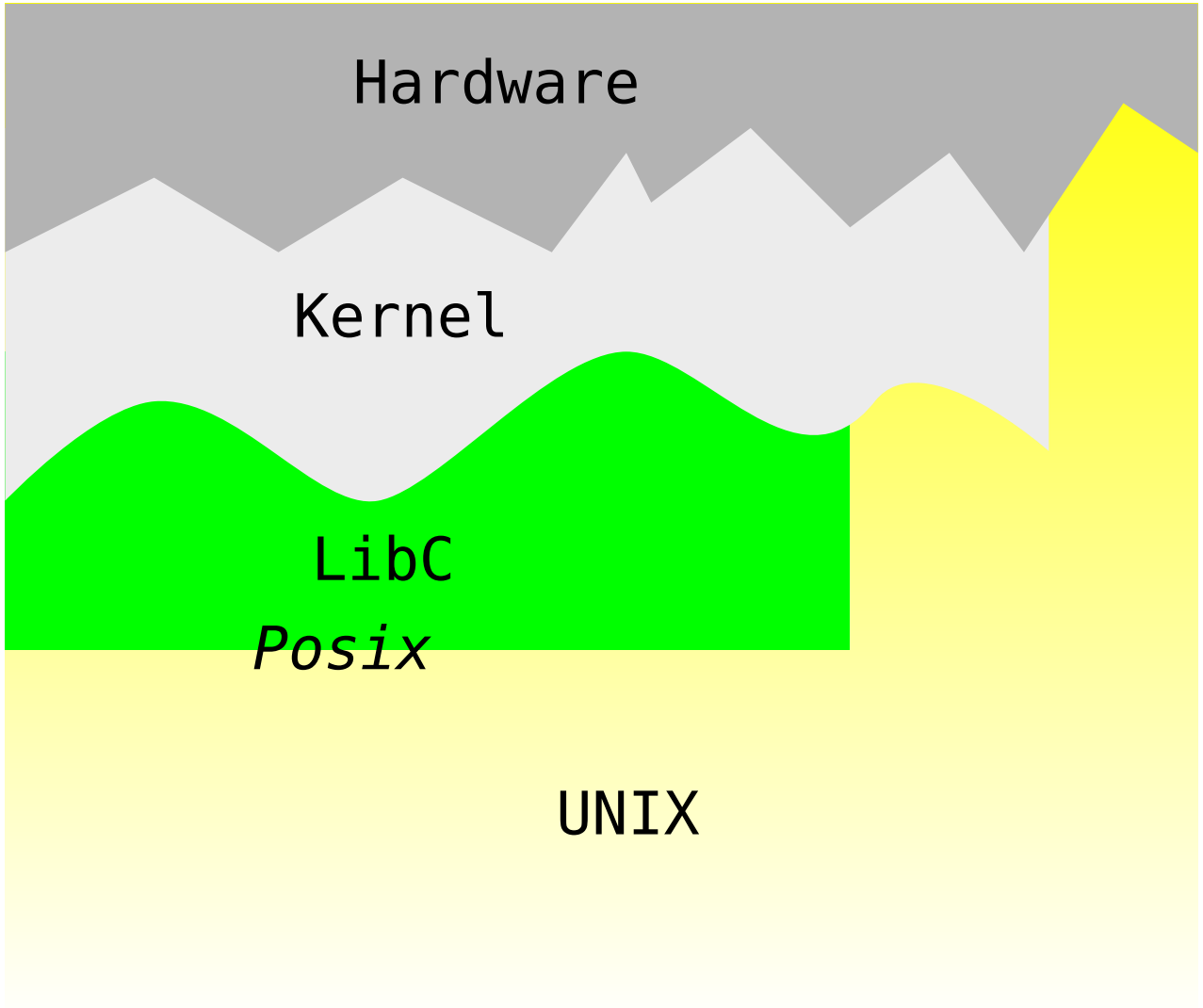
Gesucht 2 Files:

- das Kernellmage
- 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`

Mehrere Möglichkeiten

- das ganze git repository
- → nur die letzten *n* Versionen `--depth=n`
  - `git clone -b4.19 --depth=4 path_to_repository`
  - `git clone -b5.4 --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 **I**nterface: `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 **BeagleBoneGreen**
   - Partition 2 `/boot/`
4. (Re)Konfiguration
   - `sh tools/kernel.sh menuconfig`

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

Workflow

- **BeagleBoneGreen** *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

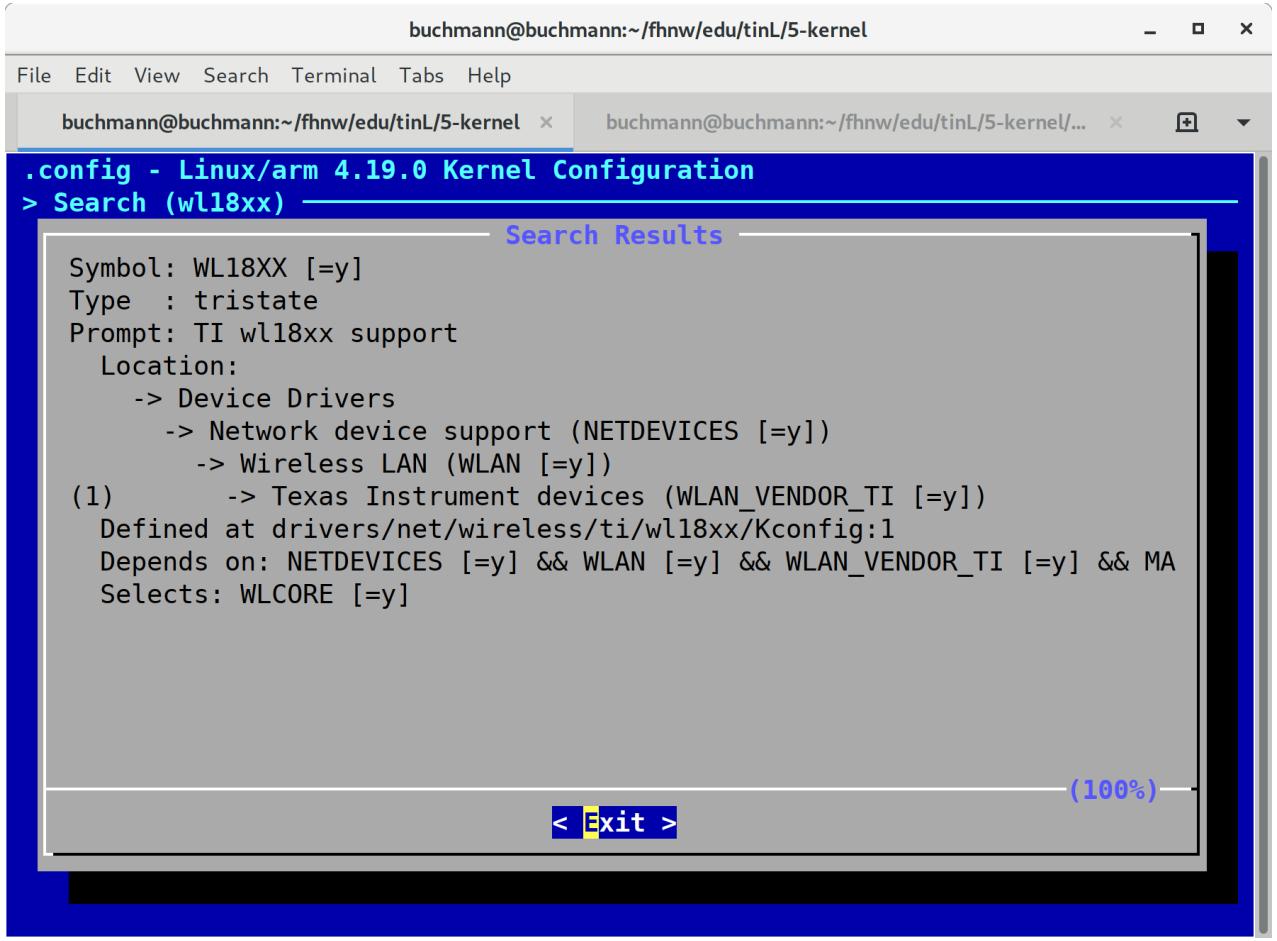
WiFi-Kernel

Konfiguration für WiFi

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

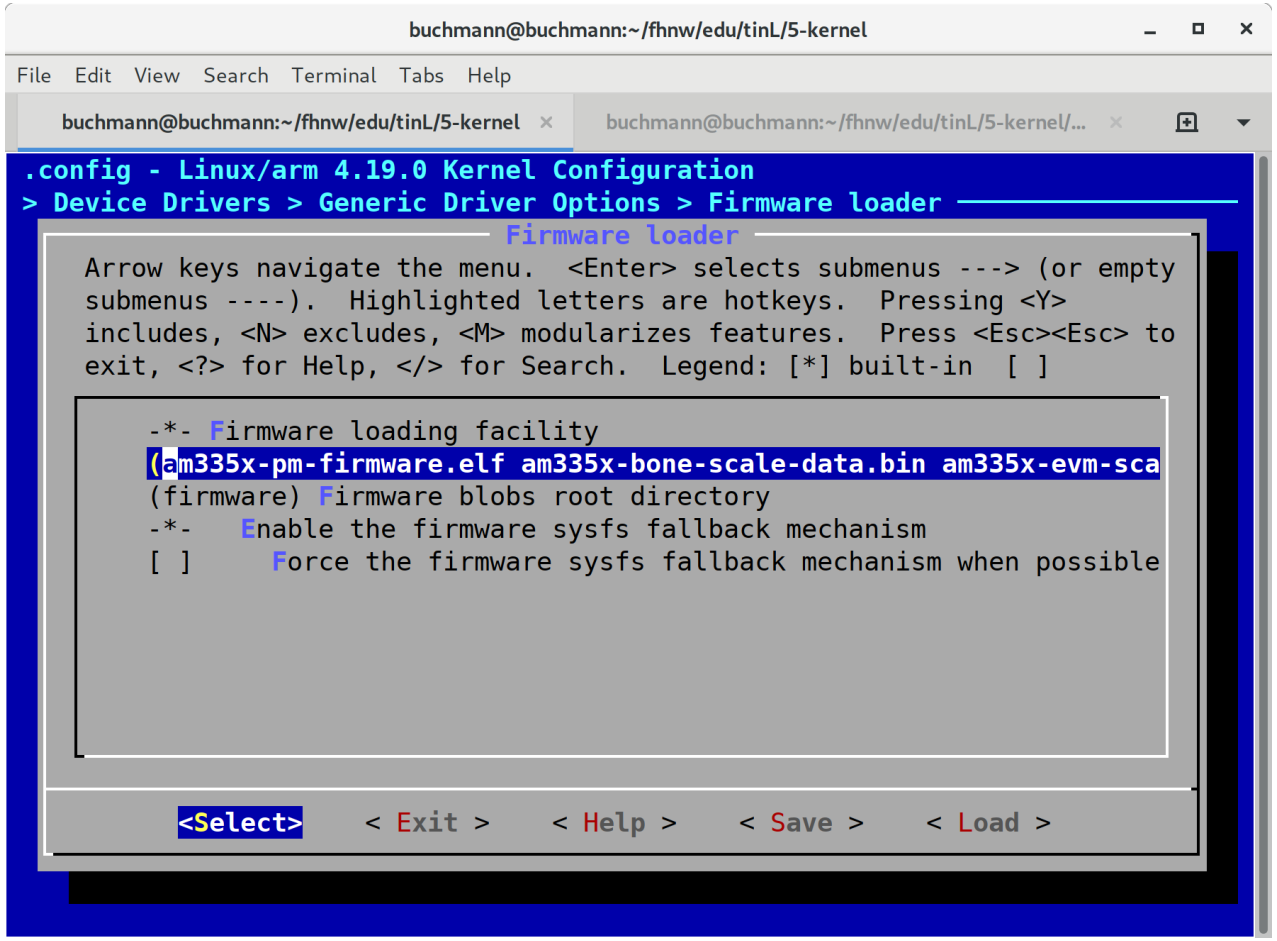
Konfiguration

Driver



Konfiguration

Firmware



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