

# Crossdevelopment

Hans Buchmann FHNW/ISE

30. April 2020

# Entwicklung von Programmen auf dem **BeagleBoneGreen**

## Nicht aus den Augen verlieren:

- ▶ alles ist ein File
  - ▶ 0 – te Näherung
  - ▶ File: *stream of bits*
  - ▶ wo sind die Files ?
- ▶ Filesysteme
  - ▶ mount
  - ▶ sshfs
- ▶ Cross development
  - ▶ *Host* ↔ **BeagleBoneGreen**

**Remark:** Keine Toolchain auf dem **BeagleBoneGreen**

# Wichtig

- ▶ wo ist was ?
  - ▶ Verzeichnisstruktur
- ▶ wo sind wir ?
  - ▶ *Host*  
oder
  - ▶ **BeagleBoneGreen**

# Ein paar Befehle

- ▶ `cat name`
  - ▶ *concatenate files and print on the standard output*
- ▶ `hexdump -C name`
  - ▶ *display file contents in hexadecimal, decimal, octal, or ascii*
- ▶ `dd if=... of=... count=...`
  - ▶ *convert and copy a file*
- ▶ `cp`
  - ▶ *copy files and directories*
- ▶ `rsync`
  - ▶ *a fast, versatile, remote (and local) file-copying tool*
- ▶ `tar`
  - ▶ *archiving utility*

# Devices sind auch Files

- ▶ SD-Karten `/dev/sdX`
- ▶ serielle Schnittstellen `/dev/ttyX`
  - ▶ `/dev/ttyUSB0` `/dev/ttyACM0`
- ▶ ...

# Crossdevelopment

- ▶ zwei Rechner
  - Host der Entwicklungsrechner
  - Target **BBG** der Zielrechner
- ▶ Development
  - ▶ Wo sind die Files
- ▶ CrossDevelopment
  - ▶ Wo sind die Files

# Outline

- ▶ Development
  - ▶ Programme auf dem *Host* für den *Host*
- ▶ CrossDevelopment
  - ▶ Programme auf dem *Host* für für den **BeagleBoneGreen**

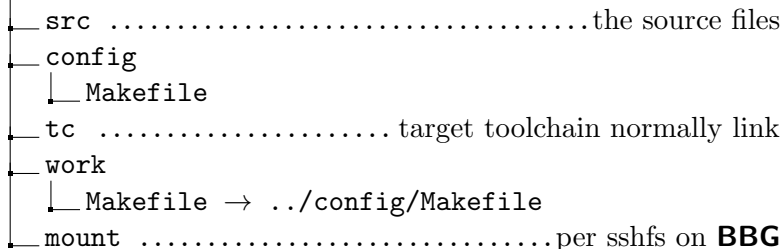
## Ziel

- ▶ CrossDevelopment (fast) gleich wie Development
- ▶ wegen **POSIX**



# Verzeichnisstruktur

## 18-crossdevelopment



# Die Komponenten

- ▶ der `Makefile`
  - ▶ steuert die Herstellung
  - ▶ das Programm `make`
- ▶ Die Sourcefiles
  - ▶ fast alle **POSIX**
  - ▶ der Compiler/Assembler/Linker

# POSIX Programme

Siehe `Makefile`

- ▶ `hello-world.c` `cpp-hello-world.cc`
  - ▶ *dynamische* vs. *statische* Libraries
  - ▶ Grösse der *executables*
- ▶ `thread-demo-1.cc`
  - ▶ viel **C++**
- ▶ `primes.cc`
  - ▶ auf dem *Host* und auf dem **BBG**
  - ▶ Ausführungszeiten

# Elementare Programme

Siehe `Makefile`

- ▶ `s-without-posix.S`
  - ▶ ein sehr einfaches GNU/Linux Assembler-Programm im *userspace*
- ▶ `c-without-posix.c`
  - ▶ ein sehr einfaches GNU/Linux C-Programm im *userspace*