# Cross Toolchain

Hans Buchmann FHNW/IME

15. Februar 2015

# Um was geht es ?

## Eigene neueste Toolchain für **RaspberryPi**

- Eigene Toolchain aus den Sourcen
- Vorteil
  - Toolchain auf den neuesten Stand
- Nachteil
  - Muss selber konfiguriert werden

# Toolchain

- die grossen zwei
  - Compiler
  - Linker
- kleinere Programme
  - Assembler
  - ...

# Toolchain
## Beispiel

- Sourcefile `source.c`
- Compilat/object File `source.o`
- Executable/Image `source`

# Cross toolchain

## 2 Verschiedene Rechner

**Host** Workstation leistungsfähiger Rechner

**Target** Eingebettetes System (**RaspberryPi**)

**Cross{Programm}** Programm (Compiler etc.) das

- läuft auf dem *Host* und erzeugt Files für das *Target*

# Cross toolchain

- erzeugt auf dem *Host* Programme für das *Target*

# GNU/Toolchain

## Zwei Komponenten

**binutils** Linker, assembler, ...

**gcc** Compiler

# Build
## die drei Schritte

- configure
- make
- make install

Remark: auf dem **Host**

# Build
## der Kontext

**prefix** wo die Toolchain auf dem *Host* installiert wird

- option `--prefix=`*`path-to-toolchain-install`*

**sysroot** wo ist das *Target* root system (auf dem *Host*)

- option
  `--with-sysroot=`*`path-to-target-sysroot`*

**target** was für eine *Target* System

- option
  `--target=armv6l-unknown-linux-gnueabihf`

Remark: Warum ???

# Verzeichnisstruktur: Host

## Build

```
12-toolchain
├── ..674c09cf-ac29-40e4-a768-5608a462a3e1 ..... tag
├── scripts
│   ├── common.sh ................. used for binutils & gcc
│   ├── binutils.sh
│   └── gcc.sh
├── build ............................... for building
│   ├── binutils
│   └── gcc
└── tc ..................................... the products
```

# Verzeichnisstruktur: Host

## Cross-development

```
12-toolchain
├── ..674c09cf-ac29-40e4-a768-5608a462a3e1 .....tag
├── src
├── tc ..................................the products
├── config
│   └── Makefile
└── work .........here we are/connected with RaspberryPi
    └── → ../config/Makefile ...................a link
```

# Test

**C** `c-source.c`

**C++** `cc-source.cc`