

Ein ganzes GNU/Linux

Hans Buchmann FHNW/IME

14. April 2015

Um was geht es ?

- ▶ ein GNU/Linux von Grund auf bauen
 - ▶ nicht meht so schwer wie auch schon
- ▶ nicht völlig automatisiert
- ▶ Alternative zu **yocto** & Co.

Ziel

GNU/Linux auf dem **RaspberryPi**

- ▶ command based
- ▶ Ethernet
- ▶ ssh
- ▶ sshfs
- ▶ moderne Toolchain inkl. *c++11* **C++**

Komponenten

RaspberryPi und *Host*

RaspberryPi

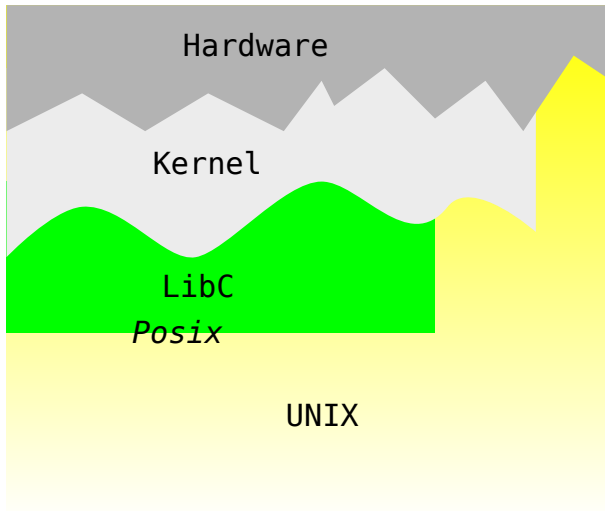
Kernel ein File

root ein Filesystem

Host

Toolchain binutils, gcc

Übersicht



Die Komponenten für RaspberryPi

Hardware **RaspberryPi**

Kernel zugeschnitten auf **RaspberryPi**

▶ <https://github.com/raspberrypi/linux.git>

root das Filesystem

LibC eglibc

▶ <http://www.eglibc.org/home>

UNIX busybox

▶ <http://www.busybox.net/>

... Weitere UNIX basierte Komponenten

▶ das configure, make, make
install Triple

Toolchain

binutils linker & Co.

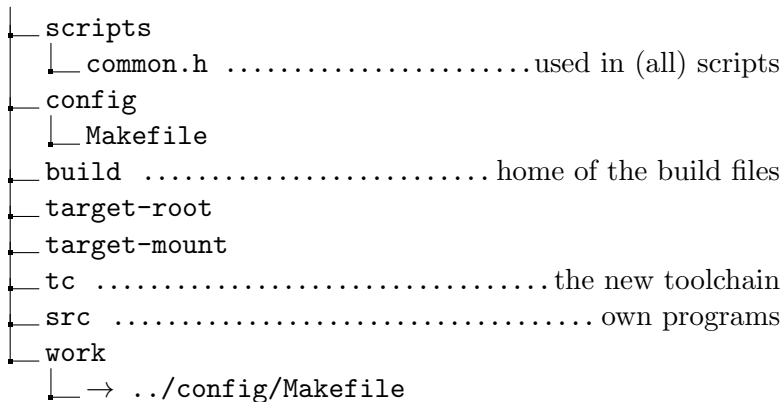
gcc compiler

- ▶ libgcc die Bibliothek für den Compiler

Remark(s):

- ▶ die Toolchain muss zweimal gebaut werden
 - ▶ für den **kernel** und libc
 - ▶ für UNIX/**POSIX**
- ▶ das target
 - ▶ cpu-vendor-os

Die Verzeichnisstruktur



Build

die Schritte 1

- ▶ binutils
- ▶ gcc-bare
- ▶ kernel mit ein paar *rules*
 - ▶ bcmrpi_defconfig
 - ▶ zImage
 - ▶ headers_install
- ▶ eglibc
- ▶ gcc
- ▶ Test
 - ▶ im Verzeichnis work

Build

die Schritte 2

- ▶ busybox
 - ▶ Installation auf SD-Card
 - ▶ fakeroot
- ▶ openssh die volle Implementation
 - ▶ zlib
 - ▶ openssl
 - ▶ openssh
- ▶ sshfs