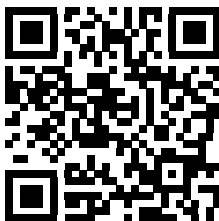


Yocto

eigene Embedded GNU/Linux Distro bauen

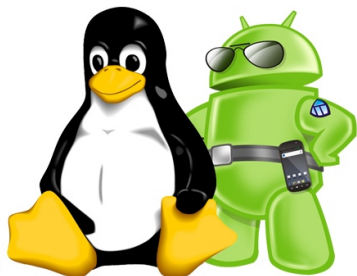
Urs Fässler; urs@bitzgi.ch

FSFE Fellowship Gruppe Zürich



www.bitzgi.ch/presentation/

8.5.2014



1

If you can't hack it, you don't own it

- Freie Geräte benötigen Freie Software
- Embedded Geräte brauchen spezielle Konfigurationen für Hardware und Software
- aufwändig,¹ ein System auf ein Gerät anzupassen

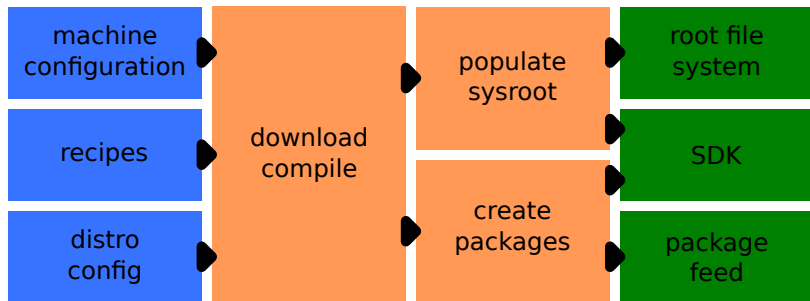
¹<http://www.androidauthority.com/linux-arm-architectures-140084/>



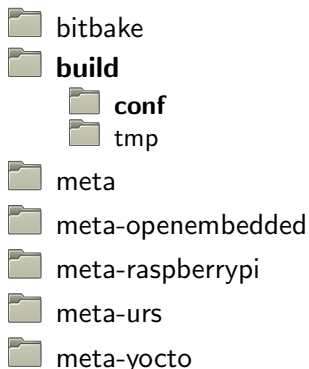
- Linux Foundation Arbeitsgruppe um Tools und Prozesse um Linux Distributionen für Embedded Geräte zu erstellen
- Breite (kommerzielle) Unterstützung

²<https://www.yoctoproject.org>

Build Übersicht



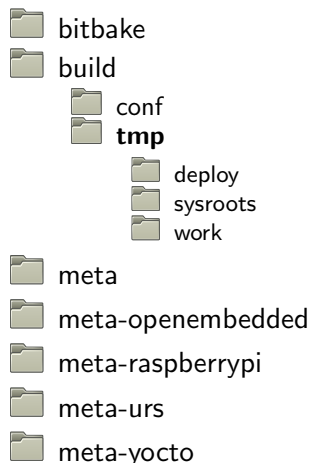
- Machine Configuration enthält angepasster Kernel sowie Softwareauswahl und globale Einstellungen
- Recipes sind in Layern organisiert und Beschreiben die Software Pakete
- Distro Config beinhaltet Einstellungen über die Feature der Distribution



- Link zu den verwendeten Layern
- Konfiguration der Build Umgebung sowie Distro Einstellungen

3

³https://commons.wikimedia.org/wiki/GNOME_Desktop_icons



- Ablage für generierte Dateien
- Pakete und Images (deploy)
- Verzeichnisstruktur wie sie auf dem Target ist (sysroots)
- Crosscompiler (sysroots)
- von Rezepten (work)

3

³https://commons.wikimedia.org/wiki/GNOME_Desktop_icons



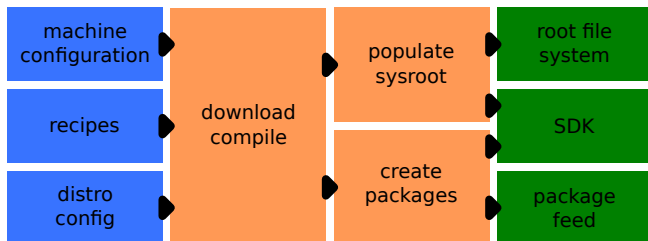
- Rezepte als *.bb Dateien
- Verzeichnisse mit Patches und Konfigurationsdateien

3

³https://commons.wikimedia.org/wiki/GNOME_Desktop_icons

```
1 SUMMARY = "A commandline OMX player for the Raspberry Pi"
2 LICENSE = "GPLv2"
3
4 DEPENDS = "libpcre libav virtual/egl boost freetype dbus"
5 RDEPENDS_${PN} += "bash procps"
6
7 SRCREV = "7af21f596378e5efecceebdff9c4a298e2d06d98"
8 SRC_URI = "git://github.com/popcornmix/omxplayer.git;protocol
9           file://0001-Remove-Makefile.include-which-includes
10
11 COMPATIBLE_MACHINE = "raspberrypi"
12
13 inherit autotools
14
15 do_install() { ... }
16
17 FILES_${PN} = "${bindir}/omxplayer* \
18               ${libdir}/omxplayer/lib*${SOLIBS}
19 ...
```


- 1 Yocto von yoctoproject.org herunterladen
- 2 `source oe-init-build-env`
- 3 herunterladen weiterer Layer (meta-raspberrypi)
- 4 Layer in `conf/bblayers.conf` eintragen
- 5 anpassen von `conf/local.conf` (`MACHINE="raspberrypi"`)
- 6 starte `bitbake rpi-basic-image`
- 7 Kaffee trinken, Mittag essen, Wochenendausflug, ...



- mit Yocto werden Pakete gebaut, nicht Software entwickelt
- `bitbake meta-toolchain-sdk` für Cross Compiler
- `bitbake <image> -c populate_sdk` für Cross Compiler mit Sysroot
- In beiden Fällen wird ein Installer Script, welches alles beinhaltet, generiert
- Die Entwickler müssen nur das Script installieren um die Software entwickeln zu können

- Hob ist ein grafisches Interface zu Bitbake
- layers.openembedded.org listet verfügbare Layer, suche nach Rezepten möglich

- spezifische Embedded GNU/Linux Distro bauen
- SDK bauen
- selbst um Updates kümmern

Präsentation und Bilder ohne Angaben: CC-BY-SA Urs Fässler
urs@bitzgi.ch