



ELBE

The Embedded Linux Build Environment

Dr. Ralf Schlatterbeck
Open Source Consulting

Email: office@runtux.com
 Web: <http://www.runtux.com>
 Tel. +43/650/621 40 17



Contents

ELBE	4
How Does It Work?	5
Packaging Sources	7
XML Overview: Native VM	8
XML Overview: Project-Section	9
XML Overview: Target-Section	10
ELBE Report	11
ELBE Command-Line	12
How to get smaller	13
ELBE Pbuilder	14



Contents

ELBE-ARM Project	15
Many Raspi Name-Alikes	16
Why Use Raspi Name-Alikes?	17
Tested Boards: Mainline	18
Archive Section	19
ELBE-ARM Makefile	20
Ressources	21



ELBE

- Build a Root-Filesystem for embedded Linux
- ... from Raspi to deeply embedded
- From a Debian-like Repository (Debian, Ubuntu, Raspbian, ...)
- Cross-Platform, e. g., build on amd64 for ARM
- Reproduceable
- Build Source-CD for license compliance
- Just re-build for Debian security upgrades
- ... in-system upgrade if supported by medium (`apt-get upgrade` on SD-Card)



How Does It Work?

- XML-Description
- Specify target image-format
- ... `.tar.gz`, SD-Card-image, UBIFS, ...
- `.tar.gz` can be used for NFS-Root
- Specify needed Debian packages
- Optional fine-tuning
- ... for creating custom config
- ... or remove unneeded stuff (doc, man)



How Does It Work?

- Everything is built in virtual machines
- QEMU for cross-platform, KVM for native VM
- Native-VM to control build process
- Cross-VM for running Debian installer
- ... and fine-tuning
- Cross-VM *inside* native VM
- Slow
- + But reproduceability counts



Packaging Sources

- ELBE creates ISO-Image
- Many software-licenses (GPL, LGPL) require sources
- Source-CD can be shipped together with product
- Think of how many embedded devices today violate licenses!
- Source-CD can be used for install
- ... to exactly reproduce a build
- Simply specify Source-CD in mirror-section of XML-File



XML Overview: Native VM

- Own section describing native VM
- ... also called `initvm`
- mirror-section specifies Debian mirror(s)
- suite (e. g. `bullseye`)
- We may specify more than one mirror
- Port forwarding to local machine
- Additional packages to install



XML Overview: Project-Section

- Specifies target architecture (buildtype)
- mirror-section specifies Debian mirror(s)
- `<suite>bookworm</suite>`
- `buildimage` section to specify inner VM
- ... e.g. additional packages
- Note: This specifies the build-VM (e.g. ARM) that runs *inside* the native VM



XML Overview: Target-Section

- Type of target (e.g. `msdoshd` for SD-Card)
- `msdoshd` refers to partition schema
- Define partitions
- `fstab` section defines mount points
- `<norecommend />`: no recommended packages
- `finetuning` section e.g. for removing unneeded stuff
- `pkg-list` section: Debian packages
- We can specify *several* targets, e.g. UBIFS *and* `.tar.gz` (e.g. for NFS-Boot filesystem)



ELBE Report

- Documents what happend during a build
- ELBE-Version and timestamp
- Package-Lists on build-VM and target
- Archive extraction before *and* after finetuning
- Log of finetuning (and error-messages)
- List of files on target image including where file came from (package name, finetuning, ...)
- Deleted files
- *Essential* for debugging!



ELBE Command-Line

- Main command `elbe` with sub-commands (like `git`)
- `elbe initvm` for creating, starting, attaching-to native VM (by default VM has 10GB)
- `elbe initvm create def.xml` Create initvm
- `elbe initvm ensure` Ensure VM runs
- `elbe initvm submit def.xml` Build image
- `elbe pbuilder create --xmlfile elbe.xml \ --writeproject elbe.prj`



How to get smaller

- For NAND-Flash use UBIFS
- Use `<norecommend />`
- Use `<diet />` mode
- ... resolves dependencies but doesn't call post-init scripts
- ... May need additional fine-tuning
- Use busybox, Debian ships two variants
- ... a statically and a dynamically linked one
- Successfully fit root image into 60MB (not GB!)



ELBE Pbuilder

- Debian package builder
- Build (local) Debian package
- For target architecture (!)
- Usual `debian` directory
- With `changelog`, `compat`, `control`, `rules`
- Built package can be fetched from VM
- ... or directly be used by built image



ELBE-ARM Project

- Uses ELBE for building images for Allwinner boards
- Build of Kernel and Bootloader (U-Boot)
- ... outside ELBE, using cross-compiler
- Configure additional packages
Using `.pkglist` files
- ELBE-XML built from configuration
- U-Boot configuration
- Device-Tree Overlays



Many Raspi Name-Alikes

- Many boards based in Chinese Allwinner CPUs
- Olimex Boards: Open Hardware very good docs
- Orange-Pi: Open Hardware
- Banana-Pi: Not open, missing documentation
- Open Hardware:
At least we can look into schematics
- Many boards have
same expansion header as Raspi
- Mainline Effort: linux-sunxi.org



Why Use Raspi Name-Alikes?

- Price
- Better Hardware-Design: e.g. Native Ethernet
- Availability: Olimex guarantees availability
- Can run Debian natively
- ... but with newer Kernel
- But not all features work with all boards (yet)
- [Status Matrix on linux-sunxi.org](#)



Tested Boards: Mainline

- Orange-Pi Zero: very cheap, no video output hardware, 256MB RAM, H2 CPU
- Banana-Pi M2+ EDU: Video not tested, 512MB RAM, H3 CPU
- Banana-Pi M2 Berry: Video not tested, 1GB RAM, V40 CPU
- Olimex OLinuXino LIME: Working Video, 512MB RAM, A20 CPU
- Olimex OLinuXino MICRO: Working Video, 1GB RAM, A20 CPU



Archive Section

- After `target` Section
- Base-64 encoded `.tar.gz` or `.bz2` Archive in XML-File
- `elbe chg_archive file.xml archive.tbz`
- e.g., Add some config-files
- ... or Script for busybox install
- Bootloader, Device tree, DT-Overlays
- Own executables
- ...
- Currently not configurable in ELBE-ARM



ELBE-ARM Makefile

- e.g. `make TARGET=orangepi_zero`
- Target is name in U-Boot
- Config in U-Boot contains name of Device-Tree file
- `make ... PKGFILES=asterisk.pkglist`
- Own Debian mirror `DEBIAN_MIRROR=localhost`
- `DEBIAN_MIRROR_PROTOCOL=http`
- Location of Linux kernel
`KERNEL=/path/to/kernel/source`
- Location of UBoot source
`BOOTLOADER=/path/to/uboot/source`



- [ELBE-ARM](#)
- [ELBE Home](#)
- [ELBE Documentation](#) with Quickstart
- [ELBE Download](#)
- [ELBE Schema Reference](#)
- [ELBE Mailinglist](#)
- There is also a developers list