Adding new SoC to OpenWrt

Hauke Mehrtens
hauke@hauke-m.de

8. October 2015

About Me

- Hauke Mehrtens
- works for Intel CHD (former Lantiq)
- OpenWrt core developer

About this talk

I will talk about

- How to integrate into upstream OpenWrt trunk

i will *not* talk about

- How to add Linux support for a SoC

OpenWrt goals

- run on 8 MB flash / 32 MB RAM
- behave like a normal Linux system
- abstract the hardware
  - e.g. same wifi configuration over all chips
- use upstream code and upstream own extensions
  - trunk normally uses a recent upstream kernel, currently 3.18 or 4.1
- image (kernel + rootfs) + optional packages
  - build one kernel binary which boots on different boards and SoCs
  - runtime board detection (e.g. device tree)
  - customization for specific boards with selected packages

Structure

- target
  - represents SoC line
  - target/linux/<target>/
- subtarget
  - represents SoC generation
  - target/linux/<target>/<subtarget>/
- profile
  - for each board
  - target/linux/<target>/<subtarget>/profiles/<vendor>.mk

target

- Target name
- CPU type
- compiler options
- kernel patches
- additional kernel files
- kernel configuration
- image building scripts
- base files

subtarget

- specialization of a target
  - different kernel options
  - different compiler options

profile

- same kernel binary as above (sub)target
- different default packages
- specific image headers

target Makefile

- target/linux/<target>/Makefile
- target name and description
- configuration for architecture, CPU type
- Linux version to use

Kernel Code

- the easiest if everything needed is upstream ;-)
- kernel patches
- kernel files
- OpenWrt kernel package

Kernel patches

- apply patches on top of the mainline kernel
- different layers
  - target/linux/generic/patches-3.18/
  - target/linux/<target>/patches-3.18/
- + when change to an existing file
- + when patch already exists
- + when backport for upstream kernel

Kernel files

- copy additional files into the mainline kernel
- same directory structure as kernel
- different layers
  - target/linux/generic/files
  - target/linux/<target>/files
- + when new driver with new file(s)

OpenWrt kernel package

- builds module as an external kernel module
- doc: `http://wiki.openwrt.org/doc/devel/packages`
- + when a tar is normally used for distribution

Kernel modules

- OpenWrt kmod package
- selected in menuconfig
- profile can select own set of kmod packages
- options should not affect kernel binary
- placed globally for all targets
  - package/kernel/linux/modules/
- placed into target directory
  - target/linux/<target>/modules.mk
- edit with: make kernel_menuconfig

Build in Configuration

- different layers
  - target/linux/generic/config-3.18
  - target/linux/<target>/config-3.18
  - target/linux/<target>/<subtarget>/config-default
- no modules selected here

File system

- spi or parallel flash
  - squashfs + appended jffs2 as overlay
- NAND flash
  - surrounded by UBIFS
- Some targets use ext4 or yaffs2, other possible

image generation

- target/linux/<target>/image/Makefile
- many vendor bootloader need a special firmware format
- also takes device tree files from /target/linux/<target>/dts/

base files

- target/linux/<target>/base-files
- default files for root file system
- etc/diag.sh
  - return status led
- etc/uci-defaults/
  - shell scripts executed when booted

Submitting patches

- submit early, submit often
  - if it boots it is sufficient
- similar rules as for the Linux kernel apply for OpenWrt

Questions?

- Hauke Mehrtens
- hauke@hauke-m.de