Building OpenWrt with the Yocto Project

Yocto Summit 2021

Tomasz Żyjewski

3MDEB
• whoami
• Who we are?
• OpenWrt description
• Advantages of using the Yocto Project
• meta-openwrt overview
• HW - short description
• Description of tests
• Boot OpenWrt community version
• Build OpenWrt with meta-openwrt
• Boot meta-openwrt image
• Summary
• Q&A
Tomasz Żyjewski
*Embedded Systems Engineer*

- `@tomzy_0`
- `tomasz.zyjewski@3mdeb.com`

- 2 years in 3mdeb
- Integration of update systems and OS creation for embedded devices
- interested in:
  - Yocto Project
  - OS updates
  - boot-time optimization
• coreboot licensed service providers since 2016
• coreboot project leadership participants
• UEFI Adopters since 2018
• Official consultants for Linux Foundation fwupd/LVFS project
• Yocto Participants and Embedded Linux experts
• Open Source Firmware enthusiasts and evangelists
Highly extensible GNU/Linux distribution for embedded devices
  - primarily used to route network traffic
Full writable filesystem with optional package management
Project started in 2004, first release in 2006
  - fun fact: releases were historically named after cocktails, such as White Russian, Kamikaze
  - naming scheme dropped after remerge of the LEDE subgroup and OpenWrt
Support over 1700 different hardware
  - for newer releases (18.06 or later) it is recommended that device has at least 16 MB of flash memory and 64 MB of RAM (128 MB of RAM is preferred)
- Why use OpenWrt?
  - extensibility - over 3000+ standardized applications, can replicate setup on any supported device
  - security - security by default, keeping software components up-to-date
  - performance and stability - each module receive a lot of testing and bug fixing
  - strong community support - OpenWrt forum and mailing lists
  - research - lots of network performance research are executed on OpenWrt systems, every successful experiments are available in OpenWrt first
  - open source - no additional cost needed, OpenWrt is provided without any monetary cost
• Releases
  ◦ current stable release: OpenWrt 19.07 (latest version 19.07.7 released on 18 Feb 2021)
  ◦ next stable release: OpenWrt 21.02 (21.02-rc1 released on April, 26th 2021)

• No stable release cycle
  ◦ breaks between releases last from a month to 20 months

• Stable version numbers are mode from the year and the month when a new stable branch was created

• Table of Hardware contains the release version that supports the device
  ◦ every release is available on the OpenWrt webpage
OpenWrt provides several thousand packages.

There is a package table which always show packages available in the latest update of the stable release.

Packages can be installed in two ways:

- via web interface and LuCI
- via command-line interface and opkg packages manager

Packages source code is also available:

- source code of some of the OpenWrt packages like LuCI or proc
d
- community repository of ported packages
- base OpenWrt repository, contains patches for basic packages

Lots of patched packages, every repository has branches for supported releases.
OpenWrt build system, known as OpenWrt Buildroot
  - based on modified Buildroot system
  - set of Makefiles and patches that automates the process of building

- Uses kconfig for the configuration of all options
- Provides an integrated cross-compiler toolchain
- Handles standard OpenWrt image build workflow: downloading, patching, configuration, compilation and packaging
- Provides a number of common fixes for known badly behaving packages
Advantages of using the Yocto Project

- Better management of the components that make up the system
  - division into layers
- Most hardware vendors provide direct support for Yocto
  - support for a new machine is easier to achieve by adding the appropriate bsp layer to the build
- Yocto Project allows to prepare more complex systems
  - extending the functionality of the system by adding layers
meta-openwrt overview

- Repository: [https://github.com/kraj/meta-openwrt](https://github.com/kraj/meta-openwrt)
- Maintainer: Khem Raj
- 3 branches: master, dunfell, hardknott
  - master branch should be compatible with gatesgarth and hardknott
    
    ```
    LAYERSERIES_COMPAT_openwrt-layer = "gatesgarth hardknott"
    ```

- 19 contributors, last contributions

![Graph showing contributions to master](image)
README.md review

- kind of outdated, not needed dependencies on meta-nodejs and meta-nodejs-contrib
- to build an image openwrt-distro-defaults needs to be add to the INHERIT variable
- consideration about TCLIBC - it is not pointed if using musl is a must
- OE release limitations - it should work with Sumo version and later

Available images

- openwrt-image-minimal - has openwrt networking and cli but no UI
- openwrt-image-base - has openwrt networking, cli, and UI (LuCI)
- openwrt-image-full - has minimal and base images functionality plus some network related packages, including relayd or tcpdump
Recipes

$ ls meta-openwrt/recipes-*/
meta-openwrt/recipes-core/:
  firewall3 fstools images iwinfo jsonpath libubox make-ext4fs netifd
  odhcp6c odhcpd packagegroups procda rpcd ubox ubus uci uclient
  ustream-ssl xtables-addons

meta-openwrt/recipes-extended/:
  images libnl-tiny liberoxml luci packagegroups ugps usbmode usign

meta-openwrt/recipes-kernel/:
  linux

meta-openwrt/recipes-networking/:
  ipset relayd uhttpd umbim umdnsd uqmi

meta-openwrt/recipes-support/:
  lua lua-socket

meta-openwrt/recipes-tweaks/:
  base-files busybox dnsip hostapd iptables modutils-initscripts packagegroups udev
Lots of core recipes use source code from [OpenWrt projects](https://github.com/openwrt) repositories.

- Each recipe has its own revision from which it retrieves the code, it should be remembered when the OpenWrt version is updated.

Installing OpenWrt specific configurations:

- Some recipes like `netifd` use also [OpenWrt Github repository](https://github.com/openwrt) to install needed scripts or configuration files.

Recipes from recipes-tweaks are fetched from default OE source, but tweaked for OpenWrt by installing additional files and scripts.

- Looks like some of them still need a lot of patches e.g. `hostapd`.
• List of provided **bbclasses**

```bash
$ tree meta-openwrt/classes/
meta-openwrt/classes/
    ├── openwrt-base-files.bbclass
    ├── openwrt.bbclass
    ├── openwrt-distro-defaults.bbclass
    ├── openwrt-kmods.bbclass
    ├── openwrt-lua.bbclass
    ├── openwrt-services.bbclass
    └── openwrt-virtual-runtimes.bbclass
```

• Inherited **openwrt-distro-defaults** sets `procd` as init manager, sets up the `DISTRO_FEATURES` and inherits another bbclasses

• **openwrt-base-files** adds OpenWrt github repository to couple of recipes (dnsmasq, hostapd, netifd, uci) and than it is used to install OpenWrt specific init scripts or configuration files
  - on master branch it use **dd3464023f18** git revision which is near v19.07.5
• Issues review
  ○ there are 12 open issues
  ○ some of them are open since 2017, some of them are couple months old
  ○ the maintainer is responsive and open to any PR improving the state of the layer
HW - short description

- Raspberry Pi 4 ver. B
  - SoC: Broadcom BCM2711
  - RAM: 4GB
  - WLAN: Cypress CYW43456
  - Ethernet: 1 Gbit port

- Powerful platform, nice for learning
  - OpenWrt releases officially support previous versions of RPi
• Goals to achieve with this proof of concept
  ○ log to the device via serial console
  ○ read system logs
  ○ connect with Ethernet cable, get IP address
  ○ log to the device via SSH
  ○ check data on LuCI
  ○ change configuration via LuCI, e.g. root password

• Basic features that allows to continue development on the system
  ○ many topics on the Internet propose to start working with the OpenWrt system by logging in via SSH or changing the configuration in LuCI
  ○ the ability to read logs allows to analyze and solve problems
The reason for the presentation and testing of meta-openwrt

- I am more familiar with Yocto than Buildroot and would prefer to set up my systems for embedded devices using the first one
- preparing an OpenWrt image with Yocto will allow to get to know most of the elements inside target system

Helping each other

- many users may come across the meta-openwrt layer in [OpenEmbedded Layer Index](#)
- raising interest in the meta layer
Boot OpenWrt community version

- For RPi 4 there is no stable version of OpenWrt released
- Can use **snapshot** version
  - community driven version, new image every few days
  - discussion in the [OpenWrt topic](#)
  - for this presentation image **rpi-4_snapshot_3.1.57-32_r16707_extra** was used
- Booting OpenWrt image allow to make comparison
Boot log

[ 0.000000] Booting Linux on physical CPU 0x0000000000 [0x410fd083]
[ 0.000000] Linux version 5.4.117 (builder@buildhost) \
(gcc version 8.4.0 (OpenWrt GCC 8.4.0 r16707-e57e460dc7)) #0 SMP Mon May 10 12:41:06 2021
[ 0.000000] Machine model: Raspberry Pi 4 Model B Rev 1.2
(...)
BusyBox v1.33.0 () built-in shell (ash)

---

rpi-dca632b6ec RaspberryPi4ModelBRev1.2 bl:Apr162020 vl805:000137ad
3.1.57-32 r16707 unknown SNAPSHOT
root:mmcblk0p2:05141719-02 boot:mmcblk0p1 cmdL:PARTUUID=05141719-02
t:2021-05-10 12:43:23 l:0.14,0.12,0.05 @up00:02:26
rootfs:944.9MB/25.1% boot:383.8MB/6.7% mem:3.7GB/59.6MB
LAN: 192.168.1.1/24(br-lan), dhcp: 192.168.1.100-249, dhcp.leases: 0
MonMay10:12:42:592021 to send to ff02::1%lan@br-lan (Address not available)
MonMay10:12:43:152021 to send to ff02::1%lan@br-lan (Address not available)
MonMay10:12:43:202021 plugin: exec_read_one: error = Cannot find device "eth1"
update-unavailable
=== WARNING! =====================================
no root password: Use the "passwd" command
=== WARNING! =====================================
root@rpi-dca632b6ec /39#
- Examine system logs
  - use logread from busybox

```
# logread -f -l 5
Mon May 10 12:42:21 2021 daemon.info urandom_seed[17614]: Seed saved (/etc/urandom.seed)
Mon May 10 12:42:22 2021 daemon.info dnsmasq[16440]: read /etc/hosts - 4 addresses
Mon May 10 12:42:22 2021 daemon.info dnsmasq[16440]: read /tmp/hosts/odhcpd - 1 addresses
Mon May 10 12:42:22 2021 daemon.info dnsmasq[16440]: read /tmp/hosts/dhcp.cfg01411c - 2 addresses
Mon May 10 12:42:22 2021 daemon.info dnsmasq-dhcp[16440]: read /etc/ethers - 0 addresses
```

- Connect to the local PC via Ethernet cable
  - get IP address right away

```
$ ifconfig eno1
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.105 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fd1d:bf:8d15:0:85ac:98c6:8e4c:89bd prefixlen 64 scopeid 0x0<global>
  inet6 fd1d:bf:8d15:0:6e9c:c0fe:e686:8c28 prefixlen 64 scopeid 0x0<global>
```
Connect to the RPi4 gateway via SSH

BusyBox v1.33.0 () built-in shell (ash)

|      |.                 .|  |  |  |.     |_   .
|  -   ||  _  |  -__|     ||  |  |  ||   _||   _
|_____||   __|_____|__|__||________||__|  |____|
|__|

rpi-dca632b6ec RaspberryPi4ModelBRev1.2 bl:Apr162020 vl805:000137ad
3.1.57-32 r16707 ssh SNAPSHOT 192.168.1.105 59768 22 /dev/pts/0
root:mmcblk0p2:05141719-02 boot:mmcblk0p1 cmdL:PARTUUID=05141719-02
t:2021-05-10 12:43:04 l:0.20,0.12,0.04 @up00:02:06
rootfs:944.9MB/25.1% boot:383.8MB/6.7% mem:3.7GB/59.9MB
LAN: 192.168.1.1/24(br-lan), dhcp: 192.168.1.100-249, dhcp.leases: 1
1620693742 00:00:00:00:00:00 0.0.0.0 tomzy-OptiPlex-7010 fdad:74b2:130f::a0f/128
MonMay1012:42:452021 plugin: exec_read_one: error = Cannot find device "eth1"
update-unavailable
=== WARNING! ===============================
o no root password: Use the "passwd" command
== WARNING! ================================
root@rpi-dca632b6ec /38#
- Open LuCi in web browser
• Change root password via LuCI
Needed layers to build meta-openwrt based image for Raspberry Pi 4

- meta-openwrt
- poky
- meta-raspberrypi
- meta-openembedded: meta-oe, meta-python and meta-networking

Used refspeccs

poky:
url: https://git.yoctoproject.org/git/poky
refspec: 6a751048e50c00261d99c2d8d69534f7a8da38a9

meta-openembedded:
url: https://git.openembedded.org/meta-openembedded
refspec: f3f7a5f1a4713f145107bb043e0d14cb3a51c62f

meta-openwrt:
url: https://github.com/kraj/meta-openwrt.git
refspec: 6e8159a07ce8991cb6b04e3cb15f82b9eadad1e5

meta-raspberrypi:
url: git://git.yoctoproject.org/meta-raspberrypi
refspec: 3ae135e590e375c8da26b003bda41c18fb977ae1
According to the documentation

- TCLIBC should point to musl
- openwrt-distro-defaults should be add to the INHERIT variable

Additionally the ENABLE_UART variable can be set to 1

- enable UART on RPi
- used in meta-raspberrypi

Tweaks needs to be added to local.conf

- kas layer management tool could be used to manage that
- layer management tools described at Yocto Summit 2020
• Boot log

[ 0.000000] Booting Linux on physical CPU 0x0000000000000000 [0x410fd083]
[ 0.000000] Linux version 5.10.17-v8 (oe-user@oe-host) \n(aarch64-ys-linux-musl-gcc (GCC) 10.2.0, GNU ld (GNU Binutils) 2.35.1) \n#1 SMP PREEMPT Mon Mar 1 09:44:55 UTC 2021
[ 0.000000] Machine model: Raspberry Pi 4 Model B Rev 1.2
(...)
Please press Enter to activate this console.
[ 6.340173] kmodloader: loading kernel modules from /etc/modules.d/*
[ 6.385165] NET: Registered protocol family 10
[ 6.391193] Segment Routing with IPv6
[ 6.408153] bridge: filtering via arp/ip/ip6tables is no longer available by default. \nUpdate your scripts to load br_netfilter if you need this.
[ 6.514195] xt_time: kernel timezone is -0000
[ 6.557067] kmodloader: done loading kernel modules from /etc/modules.d/*

Distro for Yocto Summit 2021 0.0.1

root@(none):/##
• Examine system logs
  ○ use logread from busybox

```
# logread -f -l 5
Failed to find log object: Not found
Failed to find log object: Not found
Failed to find log object: Not found
```

• Connect to the local PC via Ethernet cable
  ○ get IP address right away

```
$ ifconfig eno1
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet6 fe80::f0d:a471:f7ae:6289 prefixlen 64 scopeid 0x20<link>
  ether 34:17:eb:d7:a5:9f txqueuelen 1000 (Ethernet)
(...)
$ ping 192.168.1.1
ping: connect: Network is unreachable
```
Checking SSH connection is not possible

- networking not starting on the RPi /lib/netifd/mac80211.sh is missing
- IP address is not given by the gateway
- DHCP server may not work correctly on the meta-openwrt based image

Without IP address accessing LuCI is also impossible

Running network script needed lock directory under /run/lock

- adding it allows to read system logs
• System logs analyze

• dnsmasq issue

Fri Mar  9 12:35:19 2018 daemon.crit dnsmasq[1160]: Ubus not available: \ set HAVE_UBUS in src/config.h
Fri Mar  9 12:35:19 2018 daemon.crit dnsmasq[1160]: FAILED to start up

• uhttpd problems

Fri Mar  9 12:34:59 2018 daemon.notice proc.d: /etc/rc.d/S50uhttpd: \ Skipping invalid Lua prefix "/cgi-bin/luci=/usr/lib/lua/luci/sgi/uhttpd.lua"

• Missing some packages
  ○ collectd
  ○ dropbear
• OpenWrt vs meta-openwrt image comparison

<table>
<thead>
<tr>
<th>Functionality</th>
<th>OpenWrt community image</th>
<th>meta-openwrt image</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessing via console</td>
<td>working</td>
<td>working</td>
</tr>
<tr>
<td>read system logs</td>
<td>working</td>
<td>not working</td>
</tr>
<tr>
<td>receiving an IP address</td>
<td>working</td>
<td>not working</td>
</tr>
<tr>
<td>accessing via SSH</td>
<td>working</td>
<td>not working</td>
</tr>
<tr>
<td>accessing LuCI</td>
<td>working</td>
<td>not working</td>
</tr>
<tr>
<td>setting root password via LuCI</td>
<td>working</td>
<td>not working</td>
</tr>
</tbody>
</table>

• Lots of things do not work out of the box
  ○ this is a good initial set of functionality to check
  ○ solving the existing problems will allow for further development

• We will try to make improvements to meta-openwrt gradually
  ○ does not look usable for now
We are open to cooperate and discuss

- contact@3mdeb.com
- facebook.com/3mdeb
- @3mdeb_com
- linkedin.com/company/3mdeb
- https://3mdeb.com
- Book a call
- Sign up for the newsletter

Feel free to contact us if you believe we can help you in any way. We are always open to cooperate and discuss.
Q&A