



Buildroot for RISC-V

Using Buildroot to create embedded Linux systems
for 64-bit RISC-V

Mark Corbin



Copyright © 2019 Embecosm.
Freely available under a Creative Commons license.

About me



- Embedded Operating Systems Lead at Embecosm
- Career in Embedded Systems specialising in low-level devices and embedded Linux
- Developing Linux systems since 1996
- Currently the RISC-V maintainer for the Buildroot project

Presentation overview

- What is Buildroot?
- A little about RISC-V
- Comparing Buildroot with Yocto
- Adding RISC-V support to Buildroot
- Building a system with Buildroot
- On-going tasks and future enhancements

What is Buildroot?



'Buildroot is a simple, efficient and easy-to-use tool to generate embedded Linux systems through cross-compilation.' - buildroot.org

- Builds everything you need from source – cross tool chain, bootloader, kernel and root filesystem image
- Minimalist with a strong focus on simplicity
- Support for a wide range of boards and architectures
 - ARC, Arm, Arm64, C-Sky, m68k, Microblaze, MIPS, MIPS64, NIOS II, OpenRISC, PowerPC, PowerPC64, **RISC-V**, SuperH, Sparc, Sparc64, x86, x86_64, Xtensa
- Further information:
 - <https://buildroot.org/>
 - <https://bootlin.com/doc/training/buildroot/buildroot-slides.pdf>
 - <https://elinux.org/images/d/dd/Buildroot-Whats-New-ELC2018.pdf>

A little bit about RISC-V



- RISC-V (“risk-five”) is an open source Instruction Set Architecture (ISA) specification
 - Open source, royalty free
 - Simple – base ISA has < 50 instructions (estimated 1338 instructions for x86 in 2015)
 - Clean-slate design
 - Modular design with extensions, e.g. M (multiply/divide) A (atomic) F (single FP) D (double FP) C (compressed)
 - Stable – base and standard extensions are frozen
- RISC-V Foundation
 - Non-profit corporation that oversees the development and drives the adoption of the RISC-V ISA
 - Over 100 member organisations including Google, NVIDIA, NXP, Western Digital, Qualcomm, Samsung & **Embecosm**
 - Membership growing
 - Thales (November 2018), Raspberry Pi Foundation (Jan 2019)
- Further information:
 - <https://en.wikipedia.org/wiki/RISC-V>
 - <https://riscv.org>

Comparing Buildroot with Yocto

Buildroot	Yocto
Focussed on speed and simplicity	Extremely flexible and customisable
Easy to configure, easy to understand	Steep learning curve
Builds a root filesystem image	Builds a package feed
Open community - vendor neutral	Open community - governed by the Yocto Project Advisory Board
Over 2300 packages available	Over 8000 packages available
	Independent layers used to expand functionality

- Further information:

- <https://opensource.com/article/18/6/embedded-linux-build-tools>
- <https://bootlin.com/pub/conferences/2016/elc/belloni-petazzoni-buildroot-oe/belloni-petazzoni-buildroot-oe.pdf>

Adding RISC-V support to Buildroot (1)

- Goals
 - Add initial RISC-V 64-bit support to Buildroot
 - Provide a quick and easy way to evaluate and test RISC-V systems
 - Software support and hardware available for 64-bit
 - Work towards upstreaming features
 - Reduce number of existing RISC-V repositories
 - Minimise work
 - Avoid adding custom features
 - Use existing upstream code where possible

Adding RISC-V support to Buildroot (2)

- Choice of components (August 2018)
 - Target
 - **QEMU** (<https://www.qemu.org/>) - easily available, low cost, RISC-V support since 2.12
 - Consider SiFive HiFive Unleashed board for future
 - Tool chain
 - RISC-V support since gcc 7.1, require binutils > 2.30 to build a kernel
 - C library
 - Buildroot supports glibc, uclibc and musl. Only **glibc** has upstream RISC-V support (64-bit only)
 - Bootloader
 - **BBL** (RISC-V specific, but minimal work required)
 - U-Boot (widely used but requires work)
 - Kernel
 - mainline support since 4.15, but not able to boot under QEMU.
 - use the **4.15** branch from the **riscv-linux** git repository

Building a system with Buildroot: Overview

- Get the source
 - Download a stable release tarball or clone the git repository
- Configure the build
 - Uses Kconfig like the Linux kernel
 - Manually - '*make menuconfig*', '*make nconfig*', '*make xconfig*', ...
 - Automatically - use a predefined default config, e.g. '*make <target_board>_defconfig*'
- Build
 - '*make*'
 - output/images directory - filesystem tarball, filesystem binary image, kernel, bootloader image etc.
- Test/Deploy
 - Test with QEMU or deploy to target hardware

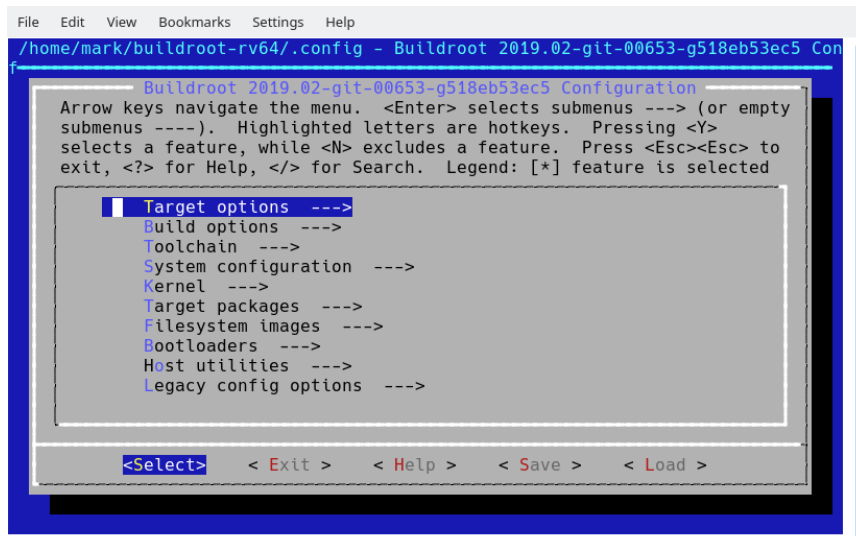
Building a system with Buildroot (1)

- Get the source
 - *git clone git://git.busybox.net/buildroot*
 - Checkout time < 30s
 - Total 136MB

```
File Edit View Bookmarks Settings Help
mark@godzilla:~$ git clone git://git.busybox.net/buildroot
Cloning into 'buildroot'...
remote: Enumerating objects: 311960, done.
remote: Counting objects: 100% (311960/311960), done.
remote: Compressing objects: 100% (94535/94535), done.
remote: Total 311960 (delta 217958), reused 309463 (delta 216171)
Receiving objects: 100% (311960/311960), 66.47 MiB | 7.81 MiB/s, done.
Resolving deltas: 100% (217958/217958), done.
mark@godzilla:~$ █
```

Building a system with Buildroot (2)

- Configure the Build
 - *'make menuconfig'* or *'make qemu_riscv64_virt_defconfig'*

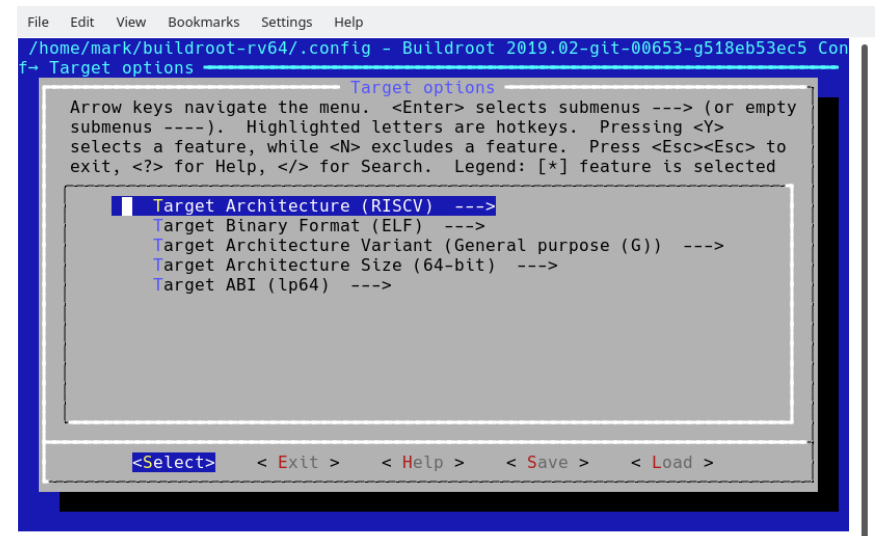


The screenshot shows a terminal window with the Buildroot configuration menu. The title bar reads "/home/mark/buildroot-rv64/.config - Buildroot 2019.02-git-00653-g518eb53ec5 Configuration". The main content area contains the following text:

```
Buildroot 2019.02-git-00653-g518eb53ec5 Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
selects a feature, while <N> excludes a feature. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] feature is selected

Target options --->
Build options --->
Toolchain --->
System configuration --->
Kernel --->
Target packages --->
Filesystem images --->
Bootloaders --->
Host utilities --->
Legacy config options --->
```

At the bottom of the terminal window, there is a navigation bar with the following options: `<Select>`, `< Exit >`, `< Help >`, `< Save >`, and `< Load >`.



The screenshot shows a terminal window with the Buildroot configuration menu. The title bar reads "/home/mark/buildroot-rv64/.config - Buildroot 2019.02-git-00653-g518eb53ec5 Configuration". The main content area contains the following text:

```
Target options --->
Target Architecture (RISC-V) --->
Target Binary Format (ELF) --->
Target Architecture Variant (General purpose (G)) --->
Target Architecture Size (64-bit) --->
Target ABI (lp64) --->
```

At the bottom of the terminal window, there is a navigation bar with the following options: `<Select>`, `< Exit >`, `< Help >`, `< Save >`, and `< Load >`.

Building a system with Buildroot (3)

- Build
 - *'make'*
 - Build time 22m 29s
 - Kernel 6.5MB
(includes bootloader)
 - Root filesystem 3.9MB
 - Disk space 7.8GB
(download 2.9GB)

```
File Edit View Bookmarks Settings Help
mark@godzilla:~/buildroot-rv64$ make
/usr/bin/make -j1 O=/home/mark/buildroot-rv64/output HOSTCC="/usr/bin/gcc" HOSTC
XX="/usr/bin/g++" syncconfig
>>> host-skeleton Extracting
>>> host-skeleton Patching
>>> host-skeleton Configuring
>>> host-skeleton Building
>>> host-skeleton Installing to host directory
>>> host-pkgconf 1.5.3 Downloading
--2019-01-13 14:14:03-- https://distfiles.dereferenced.org/pkgconf/pkgconf-1.5.
3.tar.xz
Resolving distfiles.dereferenced.org (distfiles.dereferenced.org)... 162.220.112
.102, 2602:ffdb:f80:a::102
Connecting to distfiles.dereferenced.org (distfiles.dereferenced.org)|162.220.11
2.102|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 290240 (283K) [application/octet-stream]
Saving to: '/home/mark/buildroot-rv64/output/build/.pkgconf-1.5.3.tar.xz.k8kyif/
output'

/home/mark/buildroo 100%[=====] 283.44K 615KB/s in 0.5s

2019-01-13 14:14:05 (615 KB/s) - '/home/mark/buildroot-rv64/output/build/.pkgcon
f-1.5.3.tar.xz.k8kyif/output' saved [290240/290240]
```

Building a system with Buildroot (4)

- Testing with QEMU
 - `qemu-system-riscv64 -M virt -kernel output/images/bbl -append "root=/dev/vda ro console=ttyS0" -drive file=output/images/rootfs.ext2,format=raw,id=hd0 -device virtio-blk-device,drive=hd0 -netdev user,id=net0 -device virtio-net-device,netdev=net0 -nographic`

```
File Edit View Bookmarks Settings Help
bootconsole [early0] disabled
bootconsole [early0] disabled
EXT4-fs (vda): couldn't mount as ext3 due to feature incompatibilities
EXT4-fs (vda): mounted filesystem without journal. Opts: (null)
VFS: Mounted root (ext4 filesystem) readonly on device 254:0.
devtmpfs: mounted
Freeing unused kernel memory: 136K
This architecture does not have kernel memory protection.
EXT4-fs (vda): warning: mounting unchecked fs, running e2fsck is recommended
EXT4-fs (vda): re-mounted. Opts: block_validity,delalloc,barrier,user_xattr
Starting syslogd: OK
Starting klogd: OK
Initializing random number generator... done.
Starting network: IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
udhcpd: started, v1.29.3
IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
udhcpd: sending discover
udhcpd: sending select for 10.0.2.15
udhcpd: lease of 10.0.2.15 obtained, lease time 86400
deleting routers
adding dns 10.0.2.3
OK

Welcome to Buildroot
buildroot login: █
```

On-going tasks and future enhancements

- Status of 32-bit support
 - Patches accepted into master branch January 2019
 - Requires custom glibc version
- Continuous improvement
 - Work through autobuilder results
- Migrate to upstream versions
 - kernel, 32-bit glibc.
- Add support for new features as the RISC-V software ecosystem evolves
 - U-Boot, uclibc, musl.
- Support for development boards
- Software status
 - <https://github.com/riscv/riscv-wiki/wiki/RISC-V-Software-Status>

Buildroot tests										
Date	Duration	Status	Commit ID	Submitter	Arch/Subarch	Failure reason	Libc	Static?	Data	
2019-01-16 17:57:31	24:21	OK	8d1330c	Matth Weber (U.L.E.B. Samba)	x86_64 / core2	none	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 17:56:13	08:15:41	OK	master (defb0c1)	Giulio Benetti (Microsemi of JBoss Server)	powerpc64 / power7	none	glibc	N	dir, end, log, config, defconfig	
2019-01-16 17:55:47	01:06:49	OK	master (d1330c)	Julien Boissesse (Openlab system server)	arm / cortex-m4	none	uclibc	X	dir, end, log, config, defconfig	
2019-01-16 17:55:24	01:42:14	NOK	master (d1330c)	Matth Weber (U.L.E.B. Samba)	arch64 / cortex-a72	host-glibc-v1.16.1	glibc	N	dir, end, log, config, defconfig	
2019-01-16 17:36:26	01:26:21	OK	master (d1330c)	Matth Weber (U.L.E.B. Samba)	min2	none	glibc	N	dir, end, log, config, defconfig	
2019-01-16 17:32:03	01:10:21	OK	2018-02-8-751d809	Matth Weber (U.L.E.B. Samba)	sh4	none	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 17:29:31	01:27:46	OK	master (8d1330c)	Peter Korsgaard (ps212@psd666.com, 796c-443)	microblazeel	none	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 17:25:35	03:34:18	OK	master (8d1330c)	Julien Boissesse (Openlab system server)	arm / armv2tej-s	none	glibc	N	dir, end, log, config, defconfig	
2019-01-16 17:16:32	29:44	NOK	master (d1330c)	Giulio Benetti (Microsemi of JBoss Server)	powerpc / 603e	exempt-2.4.5	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 16:45:54	01:17:11	OK	master (8d1330c)	Giulio Benetti (Microsemi of JBoss Server)	x86_64 / atom	none	musl	N	dir, end, log, config, defconfig	
2019-01-16 16:40:07	02:43:39	OK	master (8d1330c)	Julien Boissesse (Openlab system server)	mpisol / mips32	none	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 16:11:04	01:48:17	NOK	master (8d1330c)	Matth Weber (U.L.E.B. Samba)	arm / cortex-a8	host-glibc-v1.16.1	glibc	N	dir, end, log, config, defconfig	
2019-01-16 16:09:25	11:51	NOK	master (8d1330c)	André Hentschel	x86_64 / atom	ncr-8.42	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 16:08:11	01:48:06	OK	master (8d1330c)	Matth Weber (U.L.E.B. Samba)	mpisol / mips32	none	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 16:00:05	01:10:16	OK	master (8d1330c)	Yann E. MORIN	mf68 / 5200	none	uclibc	X	dir, end, log, config, defconfig	
2019-01-16 15:58:13	57:22	OK	2018-02-8-751d809	Peter Korsgaard (ps212@psd666.com, 796c-443)	microblazeel	none	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 15:56:18	01:37:55	OK	master (8d1330c)	André Hentschel	arch64_be / cortex-a53	none	glibc	N	dir, end, log, config, defconfig	
2019-01-16 15:55:28	51:40	OK	master (8d1330c)	Xoqium (Arch Linux)	arm / armv2tej-s	none	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 15:52:28	38:55	TIMEOUT	master (8d1330c)	Giulio Benetti (Microsemi of JBoss Server)	microblazeel	postage01	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 15:27:12	28:51	OK	master (8d1330c)	Giulio Benetti (Microsemi of JBoss Server)	arm / armv2tej-s	none	glibc	N	dir, end, log, config, defconfig	
2019-01-16 15:23:55	01:12:00	OK	master (8d1330c)	Yann E. MORIN	mpisol64 / mips64el	none	glibc	N	dir, end, log, config, defconfig	
2019-01-16 15:23:03	01:40:00	NOK	master (8d1330c)	Matth Weber (U.L.E.B. Samba)	powerpc / e500mc	firmware-3.4.3	uclibc	N	dir, end, log, config, defconfig	
2019-01-16 15:07:51	01:39:09	OK	2018-02-8-751d809	Yann E. MORIN	mpisol64 / mips64	none	glibc	N	dir, end, log, config, defconfig	
2019-01-16 15:02:15	02:51:56	NOK	master (8d1330c)	Xoqium (Arch Linux)	riscv64	php-7.3.0	glibc	N	dir, end, log, config, defconfig	



Thank You

mark.corbin@embecosm.com

<https://www.embecosm.com>



Copyright © 2019 Embecosm.
Freely available under a Creative Commons license.