BuildRoot
Making Embedded Linux Easy

Embedded Linux build system

Simple
Fast
Efficient

Fast and small

System description	Full build time	Uncompressed target filesystem size
Default configuration uClibc toolchain built from scratch, Busybox, tarball image.	12 minutes	1.5 MB
Default config with glibc external toolchain Busybox-only system, with a Linaro pre-built glibc toolchain.	32 seconds	3.1 MB
Real system for a product Linux kernel, glibc Sourcery toolchain, Busybox, Dropbear, Qt (no GUI), QextSerialPort, zlib, libxml2, logrotate, pppd, strace, popt, Qt application, generated as a JFFS2 image.	11 minutes	11 MB

Build times from scratch (no cache), download time excluded.

Embedded Linux build system

Simple
Fast
Efficient

Simple to use

```
$ make menuconfig

$ make

$ ls output/images/
rootfs.tar uImage u-boot.bin MLO
```

Embedded Linux build system

Simple
Fast
Efficient

Vendor neutral

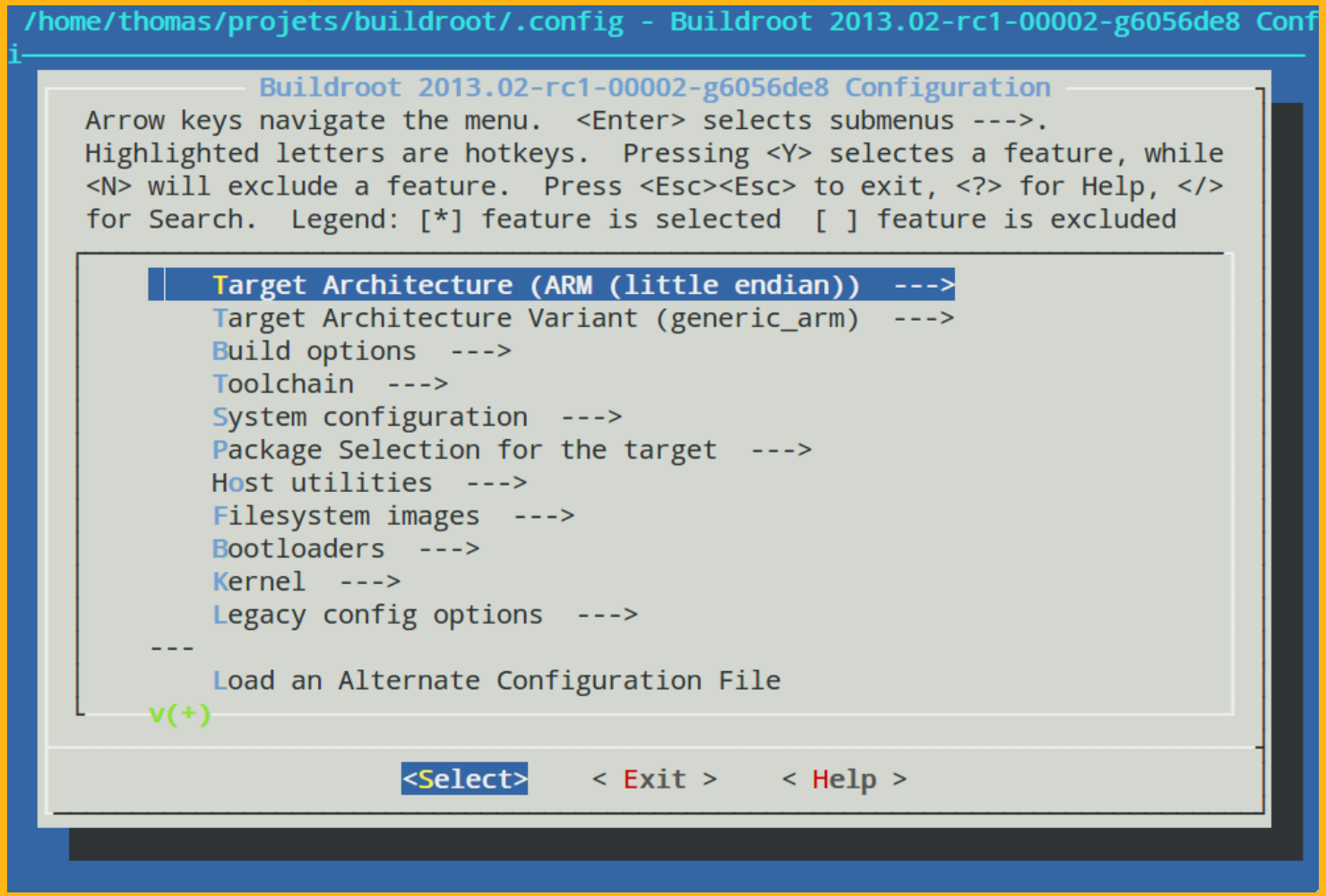
- Not controlled by any company
- Created by a pure open-source community
- Variety of actors
  - consulting company engineers
  - product manufacturers
  - hobbyists
- GPLv2 licensed

Buildroot
Making Embedded Linux Easy

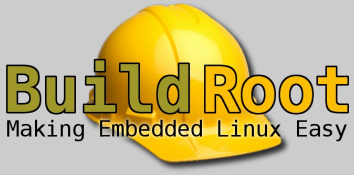
Embedded Linux build system

Simple
Fast
Efficient

Easy to configure



menuconfig / xconfig
Identical to the Linux kernel configuration



Embedded Linux build system

Simple
Fast
Efficient

Wide architecture support

- ARM
- x86 / x86-64
- PowerPC
- SuperH
- Blackfin
- Microblaze
- AArch64
- AVR32
- MIPS
- Sparc
- Xtensa

Both MMU-capable and MMU-less architectures are supported.

Buildroot
Making Embedded Linux Easy

Embedded Linux build system

Simple
Fast
Efficient

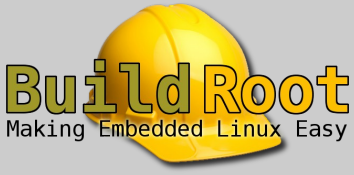
Automated build testing

Buildroot build tests

Date	Status	Commit ID	Submitter	Arch	Failure reason	Data
2013-02-13 22:09:56	OK	6056de89	Thomas Petazzoni *(Free Electrons build server)*	sh4a	none	dir, end log, full log, config, defconfig
2013-02-13 21:59:19	OK	6056de89	Thomas Petazzoni *(Free Electrons build server)*	arm	none	dir, end log, full log, config, defconfig
2013-02-13 21:48:04	OK	6056de89	Peter Korsgaard *(gcc110)*	i686	none	dir, end log, full log, config, defconfig
2013-02-13 21:34:13	NOK	6056de89	Thomas Petazzoni *(Free Electrons build server)*	arm	matchbox-lib-1.9	dir, end log, full log, config, defconfig
2013-02-13 21:26:51	OK	6056de89	Peter Korsgaard *(gcc14)*	powerpc	none	dir, end log, full log, config, defconfig
2013-02-13 21:05:16	OK	6056de89	Peter Korsgaard *(gcc110)*	i686	none	dir, end log, full log, config, defconfig
2013-02-13 21:02:13	OK	6056de89	Thomas Petazzoni *(Free Electrons build server)*	i686	none	dir, end log, full log, config, defconfig
2013-02-13 20:49:57	NOK	6056de89	Thomas Petazzoni *(Free Electrons build server)*	arm	xlib_libpthread-stubs-0.3	dir, end log, full log, config, defconfig
2013-02-13 20:48:53	OK	6056de89	Thomas Petazzoni *(Free Electrons build server)*	arm	none	dir, end log, full log, config, defconfig
2013-02-13 20:35:19	OK	6056de89	Thomas Petazzoni *(Free Electrons build server)*	arm	none	dir, end log, full log, config, defconfig
2013-02-13 20:09:54	OK	6056de89	Thomas Petazzoni *(Free Electrons build server)*	powerpc	none	dir, end log, full log, config, defconfig

Build of random configurations for many architectures, running 24/7 since one year. About 150 configurations tested every day.

Allows significant improvement of Buildroot quality by detecting problematic configurations.
http://autobuild.buildroot.net



Embedded Linux build system

Simple
Fast
Efficient

900+ packages

uClibc/glibc/eglibc toolchain, Busybox, Linux, X.org, Qt, Gtk, EFL, Gstreamer, systemd, connman, Linux, UBoot, Barebox, LTTng, Python, Perl, PHP, Lua, Xenomai, and many, many more...

Embedded Linux build system

Simple
Fast
Efficient

Easy to understand

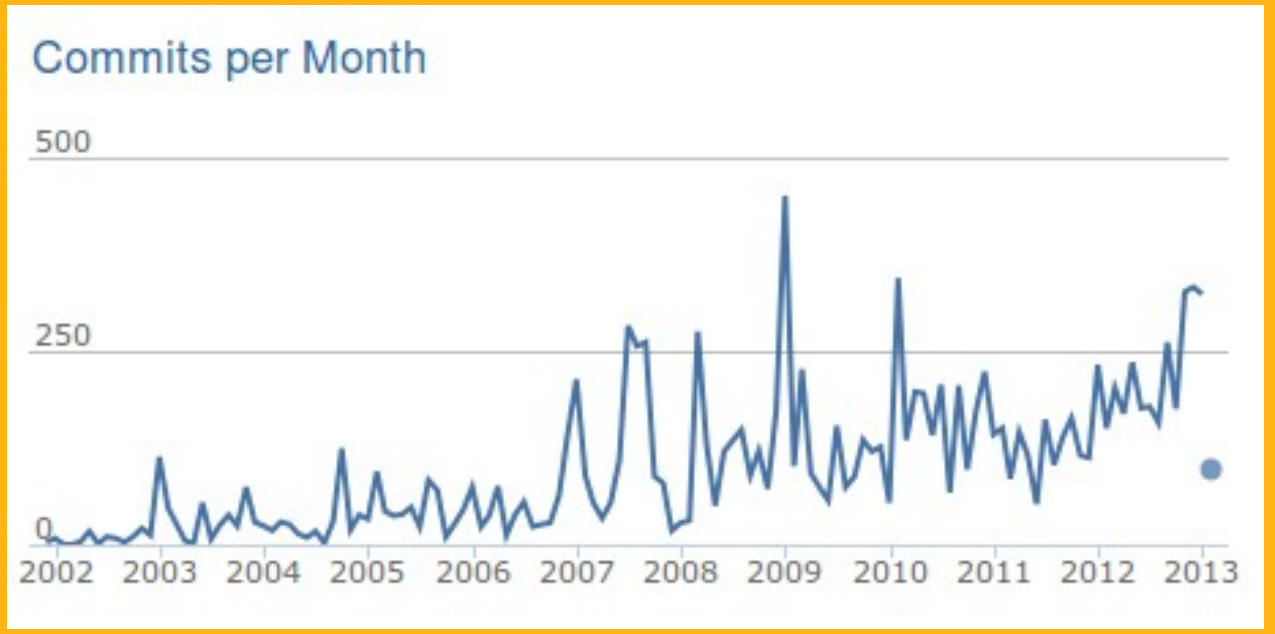
Fully written in the well-known *make* language.

The core infrastructure weighs in at approximately 2000 lines of code, comments included.

Embedded Linux build system

Simple
Fast
Efficient

Active community



Easy to add new packages

Embedded Linux build system

Simple
Fast
Efficient

package/libvorbis/Config.in

```
config BR2_PACKAGE_LIBVORBIS
        bool "libvorbis"
        select BR2_PACKAGE_LIBOGG
        help
          Library for the Vorbis open source audio decoder
          [...]
```

package/libvorbis/libvorbis.mk

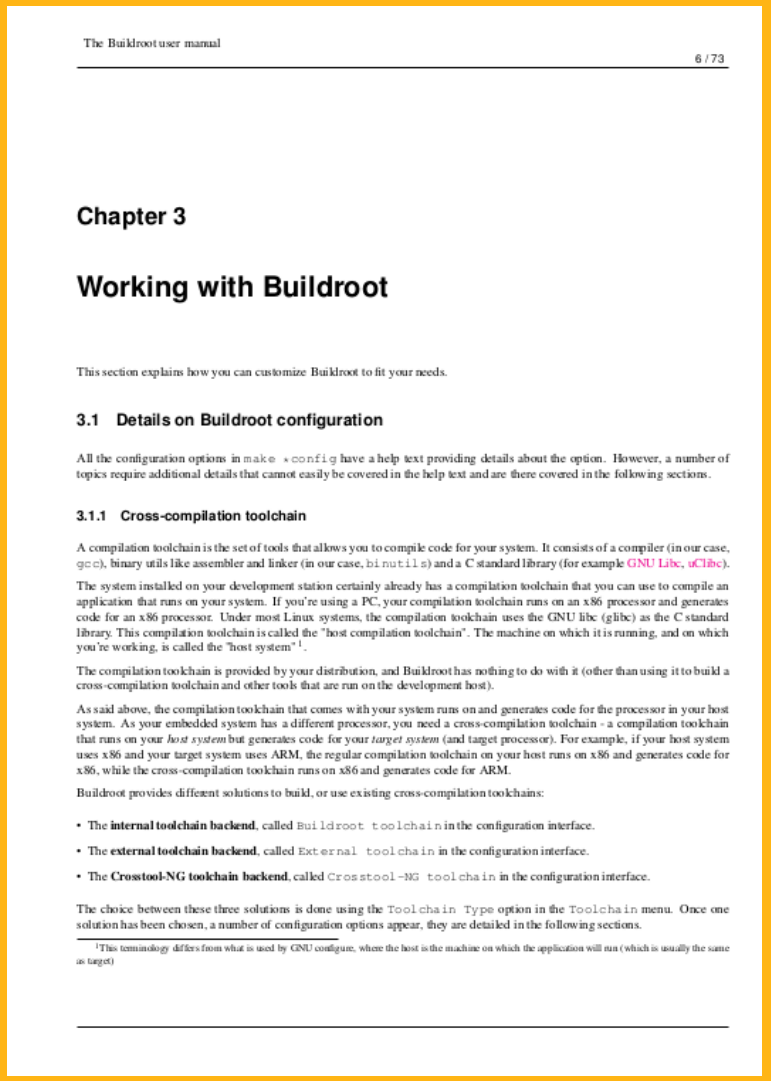
```
LIBVORBIS_VERSION = 1.3.3
LIBVORBIS_SOURCE = libvorbis-$(LIBVORBIS_VERSION).tar.gz
LIBVORBIS_SITE = http://downloads.xiph.org/releases/vorbis/
LIBVORBIS_INSTALL_STAGING = YES
LIBVORBIS_CONF_OPT = --disable-oggtest
LIBVORBIS_DEPENDENCIES = host-pkgconf libogg
LIBVORBIS_LICENSE = BSD-3c
LIBVORBIS_LICENSE_FILES = COPYING

$(eval $(autotools-package))
```

BuildRoot
Making Embedded Linux Easy

Embedded Linux build system

Simple
Fast
Efficient

Well documented



Chapter 3

Working with Buildroot

This section explains how you can customize Buildroot to fit your needs.

3.1 Details on Buildroot configuration

All the configuration options in `make *config` have a help text providing details about the option. However, a number of topics require additional details that cannot easily be covered in the help text and are there covered in the following sections.

3.1.1 Cross-compilation toolchain

A compilation toolchain is the set of tools that allows you to compile code for your system. It consists of a compiler (in our case, `gcc`), binary utils like assembler and linker (in our case, `binutils`) and a C standard library (for example GNU Libc, uClibc).

The system installed on your development station certainly already has a compilation toolchain that you can use to compile an application that runs on your system. If you're using a PC, your compilation toolchain runs on an x86 processor and generates code for an x86 processor. Under most Linux systems, the compilation toolchain uses the GNU libc (glibc) as the C standard library. This compilation toolchain is called the "host compilation toolchain". The machine on which it is running, and on which you're working, is called the "host system"[1].

The compilation toolchain is provided by your distribution, and Buildroot has nothing to do with it (other than using it to build a cross-compilation toolchain and other tools that are run on the development host).

As said above, the compilation toolchain that comes with your system runs on and generates code for the processor in your host system. As your embedded system has a different processor, you need a cross-compilation toolchain - a compilation toolchain that runs on your *host system* but generates code for your *target system* (and target processor). For example, if your host system uses x86 and your target system uses ARM, the regular compilation toolchain on your host runs on x86 and generates code for x86, while the cross-compilation toolchain runs on x86 and generates code for ARM.

Buildroot provides different solutions to build, or use existing cross-compilation toolchains:

- The **internal toolchain backend**, called `Buildroot toolchain` in the configuration interface.
- The **external toolchain backend**, called `External toolchain` in the configuration interface.
- The **Crosstool-NG toolchain backend**, called `Crosstool-NG toolchain` in the configuration interface.

The choice between these three solutions is done using the `Toolchain Type` option in the `Toolchain` menu. Once one solution has been chosen, a number of configuration options appear, they are detailed in the following sections.

[1] This terminology differs from what is used by GNU configure, where the host is the machine on which the application will run (which is usually the same as target)

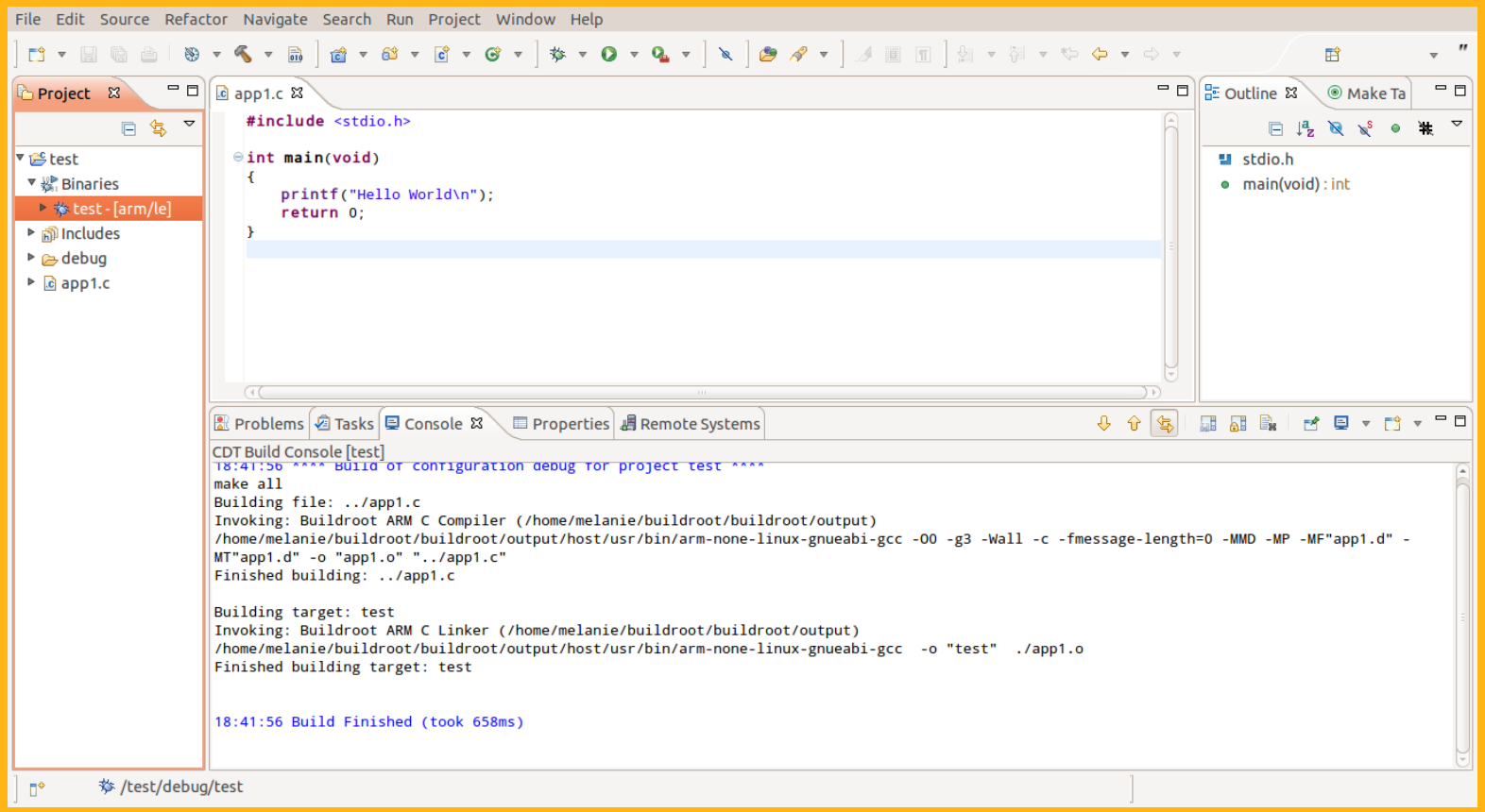
A 80 page manual detailing how to use and extend Buildroot.

Buildroot
Making Embedded Linux Easy

Embedded Linux build system

Simple
Fast
Efficient

Eclipse integration



Eclipse plugin that integrates Buildroot toolchains into Eclipse build system, enables remote execution and remote debugging.

https://github.com/mbats/eclipse-buildroot-bundle/wiki

Board support

- Pre-defined configurations for many popular boards
  - BeagleBone
  - PandaBoard
  - Rasberry Pi (in progress)
  - SheevaPlug
  - FriendlyARM Mini2440
  - QEMU emulated boards
  - AT91 evaluation kits
  - ARM Foundation v8 simulator

Embedded Linux build system

Simple
Fast
Efficient

Licensing report

- `make legal-info` automatically generates
  - a `manifest.csv` listing all components and their license
  - a `licenses.txt` with the license text of all components
  - a `licenses/` directory with all the copyright files
  - a `sources/` directory with all the sources
- Great help for open-source license compliance!

```
"PACKAGE","VERSION","LICENSE","LICENSE FILES","SOURCE ARCHIVE"
"busybox","1.21.0","GPLv2","LICENSE","busybox-1.21.0.tar.bz2"
"strace","4.7","BSD-3c","COPYRIGHT","strace-4.7.tar.xz"
"fakeroot","1.18.2","GPLv3+","COPYING","fakeroot_1.18.2.orig.tar.bz2"
"lzop","1.03","GPLv2+","COPYING","lzop-1.03.tar.gz"
"pkgconf","0.8.9","pkgconf license","COPYING","pkgconf-0.8.9.tar.bz2"
```