

LEON Buildroot

LEON Buildroot User's Manual

Table of Contents

1. Introduction	3
1.1. Overview	3
1.2. Host System Requirements	3
1.3. Installation	3
2. Quick start guide	4
3. Configuring and Building	5
3.1. Default Configurations	5
3.2. Buildroot Configuration	5
3.3. Linux Kernel Configuration	5
3.3.1. Caveats on Loosing the Kernel Configuration	5
3.4. MKLINUXIMG Configuration	6
3.5. MKPROM2 Configuration	6
3.6. Building	6
3.6.1. Rebuilding	6
4. Support	7

1. Introduction

1.1. Overview

This Buildroot release for LEON is aimed for using Buildroot as a frontend for building both the Linux kernel and a root file system with user space software. It contains default configurations for LEON systems that can be used as a baseline.

The main documentation for how to work with Buildroot is found in the regular Buildroot manual. It is provided in several different forms in the `docs/prebuilt` directory, including as text, PDF and HTML. In the top level directory there is a general Buildroot README file as well as a changelog, in the `CHANGES` file, for the upstream Buildroot release the LEON Buildroot release is based on. This manual and a changelog for the LEON Buildroot release can be found in the `docs/gaisler` directory.

This document describes the particulars of the Buildroot release for LEON. It describes what it adds in on top of the official Buildroot release it is based upon, as well as giving some pointers and tips. Our latest Linux kernel release is included as kernel patches that are automatically used when using our default configuration. Our toolchains, the MKLINUXIMG second stage bootloader as well as optionally the MKPROM2 boot loader can be downloaded automatically and configured within the Buildroot configuration interface.

This Buildroot release is not aimed to be used under the Linuxbuild LEON Linux kernel and userspace build environment, that is being phased out. It is rather aimed to be used instead of Linuxbuild.

1.2. Host System Requirements

The Buildroot frontend is only supported under Linux. See the section “System requirements” in Chapter 2 of the regular Buildroot manual for details on what is required to be installed on the host system.

In particular for configuring Buildroot, the Linux kernel and some other components it is advisable to install development versions of the Qt5 library.

1.3. Installation

Download `leon-buildroot-2022.02-1.0.tar.bz2` available from <https://gaisler.com>. Unpack it anywhere with

```
tar xf leon-buildroot-2022.02-1.0.tar.bz2
```

This will unpack to a directory `leon-buildroot-2022.02-1.0`. Enter it with:

```
cd leon-buildroot-2022.02-1.0
```

This is the top level Buildroot directory from which everything is done unless otherwise specified. Relative paths in configurations are in general relative to this Buildroot top level directory. This goes for relative paths mentioned in this manual as well.

Tools specific for LEON such as LEON Linux toolchains, MKLINUXIMG, and MKPROM2 are downloaded automatically, potentially after selecting between versions in the Buildroot configuration.

2. Quick start guide

Install Buildroot

```
tar xf leon-buildroot-2022.02-1.0.tar.bz2
```

Enter the Buildroot top level directory.

```
cd leon-buildroot-2022.02-1.0
```

Set up the default configuration:

```
make gaisler_leon_defconfig
```

Other possible default configurations are listed in Section 3.1.

Optionally, do additional configurations in the Kconfig interface:

```
make xconfig
```

See Section 1.2 for system requirements. See the configuration section printed by `make help` for additional means of configuration, e.g. `make menuconfig`.

Start the process of downloading and building everything:

```
make
```

The resulting RAM image can be found at `output/images/image.ram`. It is ready to be loaded and run in e.g. GRMON or TSIM on a LEON system with memory at address `0x40000000`. Symbols for the kernel for debugging purposes can be found in `output/images/vmlinux`. They can be loaded from within GRMON or TSIM with the **symbols** command, or in GDB with the **symbol-file** command.

Loading and running the image using GRMON3 can be done, replacing “-somedebuglink” with an appropriate debug link, with e.g.

```
grmon -somedebuglink -nosram -nb -u -e "load output/images/image.ram; symbols output/images/vmlinux; run"
```

Loading and running the image in the TSIM3 simulator can be done with e.g.

```
tsim-leon3 -nosram output/images/image.ram -sym output/images/vmlinux -e "run"
```

3. Configuring and Building

General information on how to configure Buildroot and different packages can be found in the regular Buildroot manual.

3.1. Default Configurations

This chapter lists the different default configurations for LEON systems that are provided in the release. These can be built upon to suit a particular LEON system. These default configurations also chooses a default Linux kernel configuration.

Table 3.1. Default Configurations

Configuration	Target	Notes
<code>gaisler_leon_defconfig</code>	General LEON3/4/5 systems	Assumes memory at 0x40000000
<code>gaisler_gr740_defconfig</code>	GR740	General GR740 config to base board specifics on
<code>gaisler_ut700_defconfig</code>	UT700	General UT700 config, with errata fixes enabled, to base board specifics on

As an example, to use e.g. “`gaisler_leon_defconfig`” configuration, just do

```
make gaisler_leon_defconfig
```

3.2. Buildroot Configuration

Buildroot, what packets to build and some configuration of built packets can be configured using e.g.

```
make xconfig
```

or

```
make menuconfig
```

See the regular Buildroot manual for details and alternatives.

The Buildroot configuration file is by default placed in `.config` in the top level directory. The configuration file itself is unaffected by `make clean`.

3.3. Linux Kernel Configuration

The kernel can be configured using e.g.

```
make linux-xconfig
```

or

```
make linux-menuconfig
```

See the regular Buildroot manual for details and alternatives.

3.3.1. Caveats on Loosing the Kernel Configuration

The Linux kernel configuration file is by default placed in `output/build/linux-version/.config`, where `version` corresponds to the kernel version being used. It is important to note that when doing

```
make clean
```

the Linux kernel configuration file will be removed together with everything else that is cleaned up, due to the fact that it resides in the `output/build` directory.

To save the Linux kernel configuration, configure *Buildroot* in the Kernel section, like so:

- Set the “Configuration file path” to a filename (outside of the `output` directory) where you want to save the kernel configuration. A relative path will put it relative to the Buildroot top directory.
- `BR2_LINUX_KERNEL_CUSTOM_CONFIG_FILE`
- Clear the “Additional configuration fragment files” if it is set to anything.

- BR2_LINUX_KERNEL_CONFIG_FRAGMENT_FILES

and then save the configuration to the configured destination with

```
make linux-update-config
```

for a full configuration, or

```
make linux-update-defconfig
```

for a slim configuration with only changes from default values. This procedure is not automatic, and one of the two last steps needs to be repeated after any Linux kernel configuration change for it to survive a `make clean`.

Other packages with their own `pkgname-xconfig` configuration target (or similar) can be affected in the same way. See the section on “Storing the configuration of other components” in the regular Buildroot manual for details.

3.4. MKLINUXIMG Configuration

The MKLINUXIMG second stage bootloader is enabled and configured as part of the Buildroot configuration under the bootloader section. Just as regular Buildroot packages, it is downloaded and installed automatically.

3.5. MKPROM2 Configuration

The optional MKPROM2 bootloader is enabled and configured as part of the Buildroot configuration under the bootloader section. Just as regular Buildroot packages, it is downloaded and installed automatically.

3.6. Building

When all configuration has been done, build everything with

```
make
```

The resulting images will by default be placed in the `output/images`. For example:

Table 3.2. Produced in output/images

File	Description
<code>image.ram</code>	Executable RAM file produced by MKLINUXIMG
<code>vmlinux</code>	Kernel file. Use as symbol file for kernel debug
<code>image.prom</code>	Executable ROM file from MKPROM2
<code>rootfs.cpio</code>	CPIO Root file system, suitable for initramfs.

3.6.1. Rebuilding

When adding new packages via the Buildroot configuration, or reconfiguring the kernel via the Buildroot `make` system, it is often enough to just do

```
make
```

but sometimes that is not enough. See the section “Understanding how to rebuild packages” in the regular Buildroot manual for more details. See also the section “Understanding when a full rebuild is necessary” in the regular Buildroot manual for details on rebuilding and when that is necessary.

Beware of lost Linux kernel configurations (and other configurations done via separate configuration targets) on a full rebuild. See Section 3.3.1 on how to prevent that.

4. Support

For support contact the support team at support@gaisler.com.

When contacting support, please identify yourself in full, including company affiliation and site name and address. Please identify exactly what product that is used, specifying if it is an IP core (with full name of the library distribution archive file), component, software version, compiler version, operating system version, debug tool version, simulator tool version, board version, etc.

The support service is only for paying customers with a support contract.

[Cobham Gaisler AB](#)
Kungsgatan 12
411 19 Gothenburg
Sweden
www.caes.com/Gaisler
sales@gaisler.com
T: +46 31 7758650
F: +46 31 421407

Cobham Gaisler AB, reserves the right to make changes to any products and services described herein at any time without notice. Consult the company or an authorized sales representative to verify that the information in this document is current before using this product. The company does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by the company; nor does the purchase, lease, or use of a product or service from the company convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual rights of the company or of third parties. All information is provided as is. There is no warranty that it is correct or suitable for any purpose, neither implicit nor explicit.

Copyright © 2022 Cobham Gaisler AB