

# LEON Linux

---

## LEON Linux Overview

# Table of Contents

1. Introduction .....	3
1.1. Component Summary .....	3
1.2. Following upstream stable releases .....	3
1.3. Component Compatibilities .....	3
1.4. Documentation .....	4
1.5. Download Location .....	4
1.6. Linux Kernel .....	4
1.7. Hardware Requirements .....	5
1.8. Toolchain .....	5
1.9. Buildroot distribution .....	5
1.10. Loader .....	6
1.11. LINUXBUILD build environment .....	6
1.12. Running Linux with GRMON .....	6
2. Support .....	7

# 1. Introduction

This document aims to give an overview of the current components and tools used in Linux development for the LEON architecture and how the different components are tied together.

## 1.1. Component Summary

The table below try the summarize the current LEON Linux components that Cobham Gaisler distributes patches for. Of course Linux is not limited to these components, there are lots and lots of other software available.

LINUXBUILD tries to tie some of the components below together to provide a rapid starting point for Linux development on the LEON architecture. It is recommended to set up a custom development flow that suits your particular development project. LINUXBUILD may serve as an initial setup to get started with Linux development for the LEON architecture.

Table 1.1. Component summary

Tool	Components/Options
Toolchain	<ul style="list-style-type: none"> <li>• prebuilt toolchains</li> <li>• Crosstool-NG (not part of LINUXBUILD)</li> <li>• Buildroot built toolchain</li> </ul>
Linux Kernel	<ul style="list-style-type: none"> <li>• Official Kernel Tree</li> <li>• LEON kernel Patches</li> <li>• GRLIB Driver Package (not part of LINUXBUILD)</li> </ul>
Distribution	<ul style="list-style-type: none"> <li>• Buildroot</li> </ul>
Boot Loader	<ul style="list-style-type: none"> <li>• MKLINUXIMG - LEON Linux RAM Loader</li> <li>• MKPROM2 - PROM loader</li> <li>• GRBOOT - Flight software boot loader. Not interfaced via Linuxbuild</li> <li>• U-Boot (not part of LINUXBUILD, not supported by Cobham Gaisler)</li> </ul>

## 1.2. Following upstream stable releases

Our LEON Linux kernel releases as well as our LEON Buildroot releases are set up as a relatively small number of patches on top of upstream long term stable releases. Installing our standalone releases sets by default up a git repository with a branch with our patches. The idea is that is should be easy for the user to fetch the latest updates from the upstream stable branch and rebase the corresponding LEON branch on top of the latest stable branch in order to get the latest fixes without necessarily having to wait for a new official LEON releases. Otherwise, when we do releases we do make sure to include the latest from the corresponding stable branch.

For details on following an upstream branch for the Linux kernel, see the LEON Linux User's Manual. For details on following an upstream branch for Buildroot, see the LINUXBUILD User's Manual. The instructions on following the upstream stable Buildroot branch is also relevant when not using LINUXBUILD.

## 1.3. Component Compatibilities

The LEON Linux kernel releases, the LEON Linux Buildroot releases, the LEON Linux toolchains and the MK-LINUXIMG second stage boot loader can be used in standalone fashion or integrated in various work flows. The same goes for bootloaders such as MKPROM2 and GRBOOT that is not Linux specific.

There are some implicit compatibility matters if one mixes old and new versions of the different releases. Do not use a toolchain with kernel headers from a certain kernel to build user space software and execute it under an older kernel. Building older Buildroot and kernel releases with a toolchain with a newer compiler version than what is supported upstream might not work out of the box.

The optional LINUXBUILD development environment is set up to allow for an easy starting point and integrates the various LEON Linux components into one environment. There are some extra dependencies between the LINUXBUILD version used and the LEON Linux and LEON Buildroot. In a LINUXBUILD setting, do not combine LINUXBUILD version 2.0.0, LEON Linux version 4.9-1.0, LEON Buildroot version 2017.02-1.0, and/or the sparc-gaisler-linux4.9-1.0 toolchain or later, with older versions of those components.

## 1.4. Documentation

Cobham Gaisler provides documentation for the utilities developed and various LEON specific information about the Linux kernel and usage. The documentation related to Linux is summarised in the table below.

Table 1.2. Additional Cobham Gaisler documentation

Document	Description
LEON Linux User's Manual	Documentation about configuration and usage of the LEON Linux kernel.
MKLINUXIMG User's Manual	MKLINUXIMG RAM Loader utility manual
LINUXBUILD User's Manual	LINUXBUILD build environment manual
GRLIB Driver Package Manual	Documentation about the additional GRLIB drivers distributed separately.

## 1.5. Download Location

Patches and build scripts can be downloaded from the Cobham Gaisler web server at <http://gaisler.com/anonftp/linux/linux-2.6> (even though the latest supported kernel is Linux 5.10). The official sources can be found via each project's homepage, some of the major projects used directly or indirectly are listed below.

Table 1.3. Project homepages

Tool	Homepage
LEON Linux patches and tools	<a href="http://gaisler.com/anonftp/linux/linux-2.6">www.gaisler.com/anonftp/linux/linux-2.6</a> [ <a href="http://gaisler.com/anonftp/linux/linux-2.6">http://gaisler.com/anonftp/linux/linux-2.6</a> ]
MKPROM2 product page	<a href="https://www.gaisler.com/mkprom2">https://www.gaisler.com/mkprom2</a> [ <a href="https://www.gaisler.com/mkprom2">https://www.gaisler.com/mkprom2</a> ]
GRBOOT product page	<a href="https://www.gaisler.com/grboot">https://www.gaisler.com/grboot</a> [ <a href="https://www.gaisler.com/grboot">https://www.gaisler.com/grboot</a> ]
GNU GCC compiler	<a href="http://gcc.gnu.org">gcc.gnu.org</a> [ <a href="http://gcc.gnu.org">http://gcc.gnu.org</a> ]
GNU BINUTILS	<a href="http://www.gnu.org/software/binutils">www.gnu.org/software/binutils</a> [ <a href="http://www.gnu.org/software/binutils">http://www.gnu.org/software/binutils</a> ]
Linux Kernel	<a href="http://www.kernel.org">www.kernel.org</a> [ <a href="http://www.kernel.org">http://www.kernel.org</a> ]
LTSI Linux Kernel	<a href="http://ltsi.linuxfoundation.org">ltsi.linuxfoundation.org</a> [ <a href="http://ltsi.linuxfoundation.org">http://ltsi.linuxfoundation.org</a> ]
GNU C Library (GLIBC)	<a href="http://www.gnu.org/software/libc">www.gnu.org/software/libc</a> [ <a href="http://www.gnu.org/software/libc">http://www.gnu.org/software/libc</a> ]
uClibc C Library	<a href="http://www.uclibc.org">www.uclibc.org</a> [ <a href="http://www.uclibc.org">http://www.uclibc.org</a> ]
Buildroot	<a href="http://www.buildroot.org">www.buildroot.org</a> [ <a href="http://www.buildroot.org">http://www.buildroot.org</a> ]
Busybox	<a href="http://www.busybox.net">www.busybox.net</a> [ <a href="http://www.busybox.net">http://www.busybox.net</a> ]
Crosstool-NG	<a href="http://crosstool-ng.org">crosstool-ng.org</a> [ <a href="http://crosstool-ng.org/">http://crosstool-ng.org/</a> ]
U-Boot	<a href="http://www.denx.de">www.denx.de</a> [ <a href="http://www.denx.de">http://www.denx.de</a> ], and the SPARC repository of the U-Boot project is located at <a href="http://git.denx.de/?p=u-boot/u-boot-sparc.git;a=summary">http://git.denx.de/?p=u-boot/u-boot-sparc.git;a=summary</a> [ <a href="http://git.denx.de/?p=u-boot/u-boot-sparc.git;a=summary">http://git.denx.de/?p=u-boot/u-boot-sparc.git;a=summary</a> ]

## 1.6. Linux Kernel

The current official Linux sources available at [kernel.org](http://www.kernel.org) [<http://www.kernel.org>] has support for LEON3, LEON4 and LEON5. Cobham Gaisler actively develops the Linux kernel's LEON support, submitting patches to the official kernel. Patches that hasn't reached the official kernel sources yet are provided in a separate LEON Linux kernel distribution. In this document "LEON Linux" is used to identify the official Linux kernel with the unofficial LEON patches (if any). The LEON Linux kernel distribution features back ported fixes and drivers for LEON from mainline into the longterm stable kernel version used.

When determining what LTS kernel versions to base LEON Linux releases, the selection made by the Long Term Stable Initiative (LTSI) [<http://ltsi.linuxfoundation.org>] project is taken into account when they have selected a new enough LTS kernel. The LEON Linux kernel is branched from the LTS releases from [kernel.org](http://www.kernel.org), thus it does not contain patches from LTSI but they can easily be applied when the LTS kernel version matches an LTSI version.

The GRLIB IP-cores currently supported by a vanilla official Linux kernel is listed below.

- LEON3, LEON4, LEON5 (all in single or multicore configurations)
- MMU, V8, FPU
- GPTIMER System Clock Timer
- IRQMP or IRQAMP interrupt controller
- APBUART system console
- GRETH 10/100 and 10/100/1000 Network driver using the MDIO layer of the Linux kernel
- GRPCI or GRPCI2 PCI Host support
- GRUSBHC USB 1.1/2.0 Host controller
- GRUSBDC USB 1.1/2.0 Device controller
- GRVGA Graphical Frame Buffer
- GRPS2 mouse/keyboard PS/2 controller
- GRCAN and OCCAN implements the Linux socket CAN 2.0b interface
- GRGPIO supports the generic General Purpose I/O model of Linux
- SPICTRL supports SPI master interface through the spi-fsl driver
- I2CMST support I2C master interface through the i2c-ocores driver

The LEON Linux kernel distribution available from Cobham Gaisler also includes the above drivers and the driver listed below.

- SPIMCTRL SPI master access interface

A separate Linux driver package is also distributed under the name "GRLIB Linux driver package". The intention with the driver package is to provide support for some IP-cores part of the GRLIB IP library that will not be submitted to the official kernel sources in the near future, due to different reasons. The drivers are released only for selected kernel versions. The current drivers available for GRLIB cores are listed hereafter.

- GRSPW2 DMA SpaceWire Packet kernel library
- GRSPW2 DMA SpaceWire Packet driver
- SpaceWire Router APB register configuration driver

## 1.7. Hardware Requirements

This section provides a brief listing of some of the hardware requirements for LEON Linux kernel/toolchain.

- LEON3, LEON4 or LEON5
- SPARC V8 mul/div
- MMU
- GPTIMER for system clock
- IRQMP or IRQ(A)MP for interrupt support
- D-Cache snooping
- FPU

The additional requirements that applies for *SMP* are listed below.

- Compare and swap (CASA) instruction
- Inter Processor Interrupt must have dedicated IRQ line (no IRQ sharing)
- D-Cache fast snooping

The amount of needed RAM depends on the Linux kernel configuration and on the memory needs for user space applications. It also depends whether the file system is located in RAM or not and if so how much local file storage is needed by the user space application. The default configurations for the kernel and user space that comes with Linuxbuild are fairly small and uses a RAM filesystem. They require a system with at least 32 MiB RAM.

## 1.8. Toolchain

The kernel and user application are built using a GNU GCC toolchain either prebuilt and distributed by Cobham Gaisler or a custom built toolchain. Buildroot can be used to build a custom uClibc toolchain, for embedded targets that need a small footprint LibC or for some other reason need uClibc.

## 1.9. Buildroot distribution

There are a lot of different Linux distributions available, however many of them lack support for SPARC or embedded small footprint targets. Now that both GCC and the Linux kernel have official support for the LEON

family it is easier to port distributions to LEON. Cobham Gaisler focus on maintaining support for the buildroot [<http://www.buildroot.org>] distribution.

Buildroot is a very flexible cross-compile build system which can build a major number of user-space applications. The file system image created can be used as root file system mounted via MTD from FLASH, networking NFS share or built into the kernel directly using the initramfs support (CPIO image).

## 1.10. Loader

The resulting Linux kernel (`linux/vmlinux`) must be loaded into the proper location in RAM, a basic MMU table must be setup and an OpenBoot PROM must be available that the Linux kernel request basic operations from, before the SPARC Linux kernel can successfully run.

The LEON Linux RAM Loader (**mklinuximg**) developed by Cobham Gaisler combines the virtually linked Linux kernel with a loader with the functionality described above. The LEON Linux RAM image generated by MKLIN-UXIMG can be loaded into RAM and executed by using GRMON, TSIM or GRSIM.

The LEON Linux RAM image can be booted from FLASH by using the MKPROM2 utility. It provides an efficient and fast boot FLASH/PROM procedure. It is also possible to use the commercially available flight software bootloader GRBOOT developed by Cobham Gaisler

The LEON Linux kernel can potentially also be booted using the networked bootloader U-Boot [<http://www.denx.de>] (sparc repository). This however, is not supported by Cobham Gaisler

## 1.11. LINUXBUILD build environment

Cobham Gaisler have developed a build environment named LINUXBUILD that ties together some of the above components together in one place, using make-scripts to build the separate components. Even though it is recommended to setup a custom development flow that suits your particular development project. LINUXBUILD may serve as an initial setup to get started with Linux development for the LEON architecture. Please see the LINUXBUILD documentation PDF [<http://www.gaisler.com/anonftp/linux/linux-2.6/doc/>].

The LINUXBUILD package can be downloaded from Cobham Gaisler web server in the `linux/linux-2.6/linuxbuild` [<http://gaisler.com/anonftp/linux/linux-2.6/linuxbuild>] subdirectory.

## 1.12. Running Linux with GRMON

When running Linux while being connected to the hardware with GRMON, it is important to start GRMON with the `-nb` startup option. Otherwise execution will stop on the first page fault. Page faults are a normal occurrence in Linux, and when GRMON is started with the `-nb`, the kernel gets to handle them without interference from GRMON. When GRMON is not connected, nothing needs to be done to ensure that the kernel gets to handle page faults, as that is the default behavior after reset.

## 2. Support

For support contact the Cobham Gaisler support team at [support@gaisler.com](mailto: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](http://www.caes.com/Gaisler)  
[sales@gaisler.com](mailto:sales@gaisler.com)  
T: +46 31 7758650  
F: +46 31 421407

CAES reserves the right to make changes to any products and services described herein at any time without notice. Consult CAES or an authorized sales representative to verify that the information in this document is current before using this product. CAES 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 CAES; nor does the purchase, lease, or use of a product or service from CAES convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual rights of CAES 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 © 2021 Cobham Gaisler AB