LEON Linux

USER MANUAL

RELEASED OCTOBER 2024

LEON Linux Overview

Table of Contents

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

This sections summarizes the LEON Linux components that we distribute in one form or another. Of course Linux is not limited to these components, there are lots and lots of other software available that can be built for LEON systems. The components can be used individually or together. They can be used in a custom work flow or using our LEON Buildroot environment.

We provide a LEON Buildroot distribution that builds upon the upstream Buildroot project. It ties many of the components below together in one development environment, as well as building a user environment with support for a large number of different software packages. It serves as a rapid starting point for Linux development for the LEON architecture. It can automatically download needed LEON utilities, be used to configure, develop and build both kernel and user space environment, produce RAM images with MKLINUXIMG, and create bootable ROM images from the RAM images with MKPROM.

Table 1.1. Component summary

Tool	Components/Options
Linux Kernel	• Official Kernel Tree • LEON kernel Patches • GRLIB Driver Package
Distribution	• Buildroot - distribution as well as development environment tying the other components together
Toolchain	• Prebuilt toolchains • Buildroot built toolchain
Boot Loader	• MKLINUXIMG - LEON Linux RAM Loader • MKPROM2 - PROM loader • GRBOOT - Flight software boot loader. Not interfaced via Buildroot

1.2. Documentation

We provide 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. LEON Linux Documentation distributed by us

Document	Description
LEON Linux Overview	This document giving an overview of LEON Linux components and tools.
LEON Linux User's Manual	Documentation about configuration and usage of the LEON Linux kernel.
LEON Buildroot User's Manual	User's manual for LEON Buildroot specifics.
Buildroot Manual	Upstream Buildroot manual for the versions our releases are based upon.
MKLINUXIMG User's Manual	MKLINUXIMG RAM Loader utility manual
GRLIB Driver Package Manual	Documentation about the additional GRLIB drivers distributed separately.

1.3. Download Locations

LEON Linux components distributed by us can be downloaded via https://www.gaisler.com/index.php/downloads/linux. Upstream sources can be found via each project's homepage. Some of the major projects used directly or indirectly are listed below.

Table 1.3. Download locations and project homepages

Tool	Homepage
LEON Linux product page	https://www.gaisler.com/linux
LEON Linux download page	https://www.gaisler.com/index.php/downloads/linux
MKPROM2 product page	https://www.gaisler.com/mkprom2
GRBOOT product page	https://www.gaisler.com/grboot
GNU GCC compiler	https://gcc.gnu.org
GNU BINUTILS	https://www.gnu.org/software/binutils
Linux Kernel	https://www.kernel.org
GNU C Library (GLIBC)	https://www.gnu.org/software/libc
uClibc-ng Embedded C Library	https://uclibc-ng.org
Buildroot	https://www.buildroot.org
Busybox	https://www.busybox.net
U-Boot	https://www.denx.de, and the SPARC repository of the U-Boot project is located at https://source.denx.de/u-boot/custodians/u-boot-sparc.

2. Components

2.1. Linux Kernel

The current official Linux sources available at kernel.org [https://www.kernel.org] has support for LEON3, LEON4 and LEON5. We actively develop 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) in our LEON Linux releases. Our LEON Linux kernel releases are based on upstream long-term support (LTS) kernel kernel versions. The LEON Linux kernel distribution features back ported fixes and drivers for LEON from mainline into the longterm stable kernel version used.

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 us also includes the above drivers and the driver listed below.
- SPIMCTRL SPI master access interface
- L4STAT driver for GR740
- GPTIMER watchdog driver (including support for the GR740 PLL Watchdog)
- GRCG clock gating driver
- GRWATCHDOG watchdog driver

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

2.2. Buildroot distribution

Buildroot is a flexible cross-compile build system which can build a major number of user-space applications and create root file systems for different underlying media. It can also be used a common frontend to configure and build the Linux kernel, toolchains as well as bootloaders, all from one environment. LEON Buildroot can automatically download needed LEON utilities, be used to configure, develop and build both kernel and user space environment, produce RAM images with MKLINUXIMG, and create bootable ROM images from the RAM images with MKPROM. From our LEON Buildroot 2022.02-1.x and onward this is our officially supported LEON Linux development and build environment. LEON specific documentation about Buildroot is found in the LEON Buildroot User's Manual. The general Buildroot manual is included in the LEON Buildroot release.

Buildroot targets embedded systems. It supports several hundreds of useful userspace applications which can be automatically included in the generated root file system as selected by the user. Our LEON Buildroot environment

makes it easy to get started with default configurations ready to build example RAM images with both kernel and RAM file system, and to be used as a starting point for further development. 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).

Buildroot is also our officially supported environment for NOEL Linux development for systems with our RISC-V NOEL processors.

2.3. Toolchain

The kernel and user applications are built using a GNU GCC toolchain, either a prebuilt toolchain distributed by us or a custom built toolchain. Buildroot can also be used to build custom toolchains in different configurations.

2.4. MKLINUXIMG LEON Linux RAM Loader

When building a SPARC Linux kernel, 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.

Our MKLINUXIMG LEON Linux RAM Loader combines the virtually linked Linux kernel with a loader with the functionality described above. The LEON Linux RAM image generated by MKLINUXIMG can be loaded into RAM and executed directly using GRMON or TSIM.

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 us.

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 us.

2.5. Legacy LINUXBUILD build environment

The LINUXBUILD LEON Linux build environment, developed by us, ties together some of the above components together in one place. It has from LEON Buildroot 2022.02-1.x, been replaced by LEON Buildroot as the officially supported build environment.

3. Considerations

3.1. 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. If 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.

3.2. 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 are not Linux specific. Buildroot can be used as a development environment, tying many of these components together. From LEON Buildroot 2022.02-1.0 and onwards, our Buildroot release by default downloads compatible tool versions automatically.

There are some implicit compatibility matters if one mixes old and new versions of the different releases when setting up things in a custom work flow. 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.

3.3. 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 or additions on top of upstream long term stable releases. Installing our standalone Linux kernel release sets by default up a git repository with a branch with our kernel patches applied. The idea is that is should be easy for the user to fetch the latest updates from the upstream LTS 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 from us. When we do Linux kernel and Buildroot releases we generally make sure to rebase it on the latest release of the corresponding long-term supported 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 LEON Buildroot User's Manual.

3.4. 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.

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.

Frontgrade Gaisler AB
Kungsgatan 12
411 19 Göteborg
Sweden
frontgrade.com/gaisler
sales@gaisler.com
T: +46 31 7758650
F: +46 31 421407