



Building the LINUX kernel for LEON

LINUX build process overview

Written by Konrad Eisele, Daniel Hellstrom

*LinuxBuild
Version 1.0.7
July 2014*

LINUX build process overview

Konrad Eisele, Daniel Hellstrom

Copyright © 2014 Aeroflex Gaisler AB

Table of Contents

1. LINUXBUILD	1
1.1. Introduction	1
1.2. Requirements	1
1.2.1. Installing Toolchain	1
1.3. Download Location	2
2. Installing	3
2.1. Toolchain	3
2.2. Linux kernel	3
2.3. Buildroot distribution	4
2.4. Mklinuximg Linux RAM loader	4
2.5. Mkprom	4
3. Configuring	5
3.1. Predefined configurations	6
3.2. Toolchain configuration	6
3.3. Buildroot configuration	6
3.3.1. Buildroot toolchain configuration	6
3.3.2. Network Configuration	6
3.4. Linux configuration	7
3.5. LEON Linux Loader configuration	7
3.6. MKPROM2 configuration	7
4. Building	8
5. Upgrading	9
6. Support	10

1. LINUXBUILD

1.1. Introduction

This document describes how the Linuxbuild utility is used to build one or more of the components listed below. The Linuxbuild utility consists of small Makefile-scripts for building and configuring some of the Linux tools that Aeroflex Gaisler provides patches for.

- Linux Kernel
- Buildroot user-space root file system
- Linux Kernel RAM image
- PROM/FLASH image

The Linuxbuild utility is a quick way of getting started with Linux development for the LEON architecture, it ties different components together to build a complete Linux environment. Each component is designed to be used separately from each other, one can see Linuxbuild as an example utility that provides a quick way of getting started with Linux development using the different tools and components. Currently the following components are supported in Linuxbuild.

- Linux Kernel + LEON Linux patches
- LEON Linux RAM loader (**mklinuximg**)
- Buildroot + LEON patches
- MKPROM2

Settings for standard LEON Linux configurations are available within the Linuxbuild package and can also be created by the user. Predefined configurations can be found in the `gaisler/configs` directory.

Note also that since each component is configured separately it is sometimes needed to set the same configuration option in multiple locations.

1.2. Requirements

- SPARC/LEON Linux Toolchain (Buildroot can be used to build a toolchain)
- MKPROM2 - for creating PROM/FLASH images
- wget
- git
- Internet access

Buildroot requires a number of tools such as bison, flex, msgfmt, makeinfo, etc. please see respective tool's homepage for requirements.

1.2.1. Installing Toolchain

Unless a custom toolchain is built with `crosstool-ng` or the Buildroot tool, the standard SPARC/LEON Linux toolchain should be installed before proceeding.

The GCC-4.4.2 multilib based toolchain is downloaded from the Aeroflex Gaisler web server at <http://gaisler.com/anonftp/linux/linux-2.6/toolchains/sparc-linux-4.4.2/>.

The toolchain is installed into the `/opt` directory creating the resulting directory `/opt/sparc-linux-x.y.z-toolchains/multilib`. The `bin` directory containing `sparc-linux-gcc` should be added to the `PATH` variable:

```
$ export PATH=/opt/sparc-linux-4.4.2-toolchains/multilib/bin:$PATH
$ which sparc-linux-gcc
/opt/sparc-linux-4.4.2-toolchains/multilib/bin/sparc-linux-gcc
```

Note that the toolchain path is hardcoded and cannot be installed to another directory.

1.3. Download Location

Table 1.1. Build steps

Tool	Download location
LINUXBUILD	http://gaisler.com/anonftp/linux/linux-2.6/linuxbuild/linuxbuild-x.y.z.tar.bz2
Linux toolchain	http://gaisler.com/anonftp/linux/linux-2.6/toolchains/sparc-linux-4.4.2/
MKPROM2	http://gaisler.com/anonftp/mkprom2/linux/mkprom2-2.0.36.tar.gz

2. Installing

After downloading the Linuxbuild package it is extracted using **tar -xf**.

```
$ tar -xf linuxbuild-x.y.z.tar.bz2
$ cd linuxbuild-x.y.z
```

2.1. Toolchain

Before configuring and using Linuxbuild the SPARC/LEON Linux toolchain must be installed, unless Buildroot is used to build a uClibc toolchain. See Section 1.2.1.

2.2. Linux kernel

A Linux kernel package then needs to be installed the first time in the KConfig GUI.

```
$ make xconfig
... do SELECT/INSTALL LINUX...
```

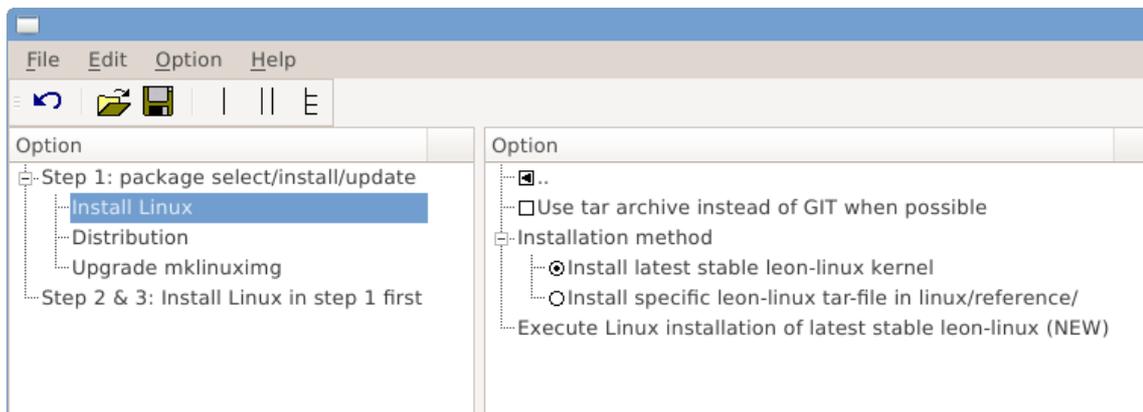


Figure 2.1. Selecting Linux components for installing/upgrading

By default, Linux is installed by cloning the official Linux GIT repository and applying patches therein. If the option "Use tar archive instead of GIT when possible" is selected, a tar archive from kernel.org is downloaded and used as the patch-base.

By default, the latest stable LEON Linux kernel distribution from Aeroflex Gaisler is automatically downloaded when installing Linux. To install a different LEON Linux kernel distribution, download a LEON Linux kernel package from the Aeroflex Gaisler website and place it in the `linux/reference` directory, before starting the GUI. Then choose "Install specific leon-linux tar-file".

The installation is done by executing the install command from within the GUI. When double-clicking on "Execute linux upgrade" a dialog will pop up asking weather to exeute the installation in a new xterm or in the parent shell:

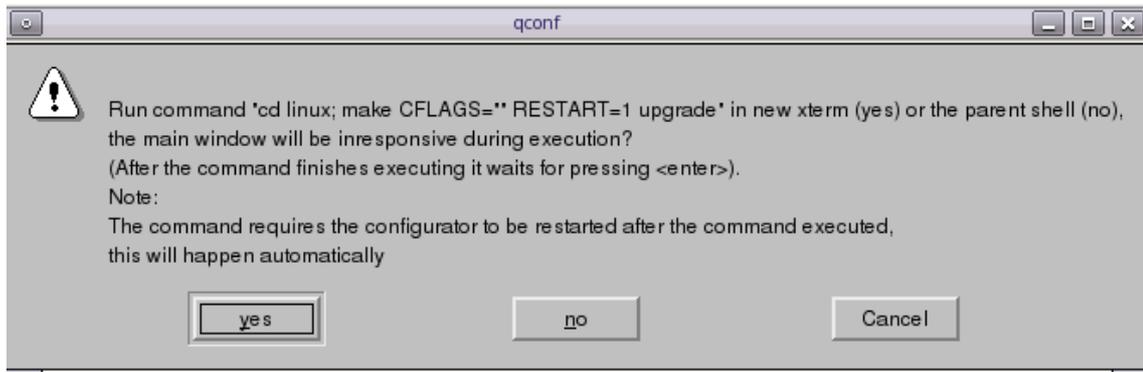


Figure 2.2. Dialog asking weather to execute a command from within the GUI

When installing, upgrading and and permanently loading ans saving a configurations, the configurator is restarted automatically so that the configuration change is shown in the configurator.

2.3. Buildroot distribution

The buildroot distribution is preinstalled. In Step 1 in the GUI, one can choose to use no distribution, e.g. to instead provide Linux with a ready made file system image through the Linux configuration.

2.4. Mklinuximg Linux RAM loader

The latest mklinuximg at the time of the release of Linuxbuild is preinstalled. This package can be upgraded in Step 1 in the GUI.

2.5. Mkprom

The mkprom utility is downloaded and installed manually separately from Linuxbuild.

3. Configuring

After the selected components have been downloaded or upgraded, Linuxbuild and each selected component is configured using one of the following make targets (**make xconfig**):

- This is the recommended interface
- Not recommended, but provided as is
- xconfig - Qt based GUI (Qt-3/4 libs required)
- gconfig - GTK based GUI
- menuconfig - ncurses based terminal interface

The buildroot and Linux configuration trees are displayed as subtrees inside the main screen:

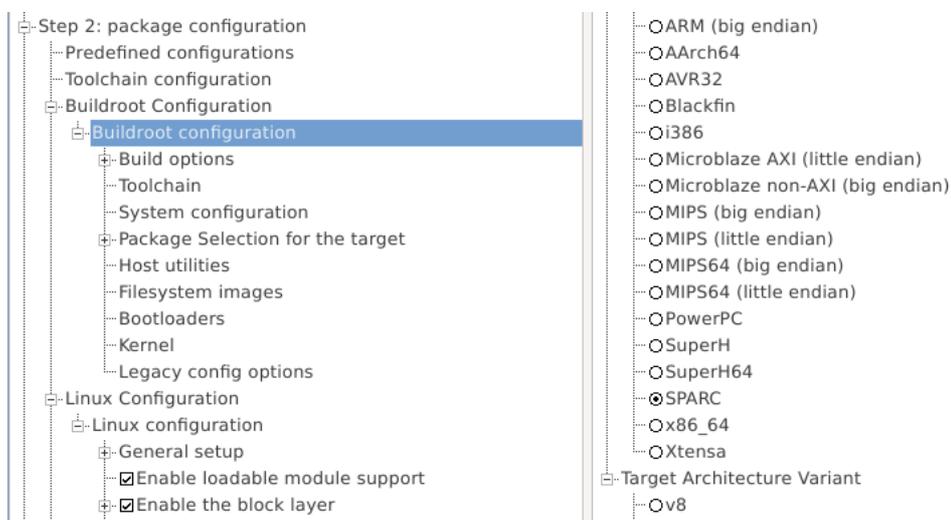


Figure 3.1. The buildroot configuration tree inside the main configuration tree

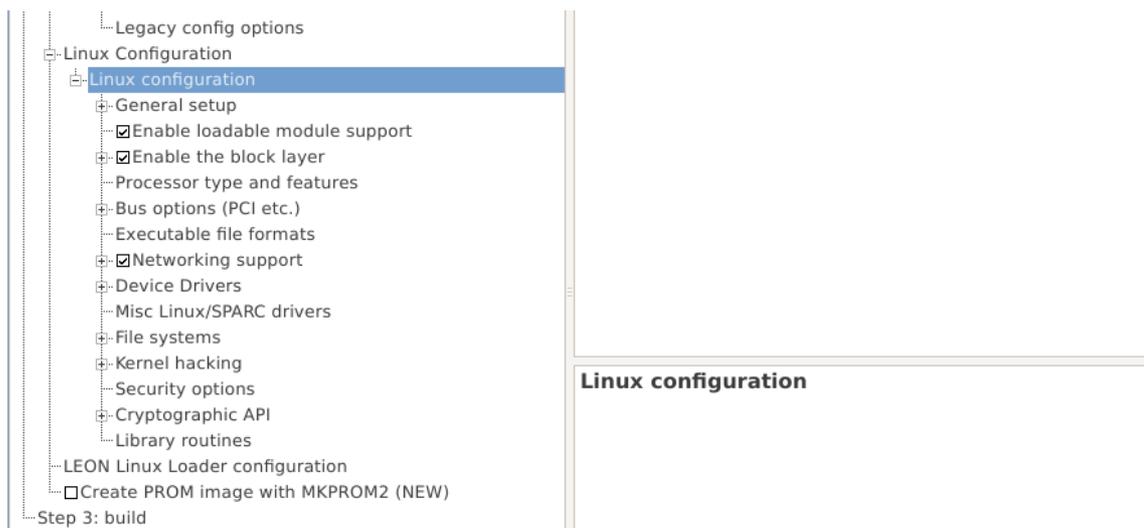


Figure 3.2. The LINUX configuration tree inside the main configuration tree

Note that since each component is configured separately it is sometimes needed to set the same configuration option more than once in different configuration GUIs.

3.1. Predefined configurations

Prepared LEON configurations can be found in `gaisler/configs`. They can be loaded by Linuxbuild under "Predefined configurations" in the GUI. In addition, there the current configuration can be saved back into a predefined configuration or saved as a new predefined configuration.

To load a predefined configuration, select an available predefined configuration and double click "Load the selected configuration". The GUI will restart after loading the configuration. Note that loading a configuration overwrites the current configuration of all components.

To save the current configuration back to a predefined configuration, select an available predefined configuration and double click "Save the current configuration back...". To save the current configuration to a new configuration, configure a name for the new configuration (`LB_SELECTED_SAVE_CONFIG`) and double click "Save the current configuration into a new...". The GUI will restart after saving the configuration.

3.2. Toolchain configuration

Here, a choice can be made to use an external toolchain in path (default), an external user provided toolchain or using a Buildroot toolchain. The choice here must correlate to the Buildroot toolchain configuration.

3.3. Buildroot configuration

Buildroot is used to build user-space applications and tool chains. It is in the Buildroot configuration the decision is made whether to compile for systems with FPU (`hfleonv8`) or to be compatible for systems without FPU (`sfleonv8`). There is no support to compile for SPARC v7. Mainline support has been dropped for both Linux and Buildroot.

3.3.1. Buildroot toolchain configuration

By default, an external toolchain is used. Aeroflex Gaisler does not distribute uClibc tool chains at the time of writing. In order to build such a toolchain, Buildroot can be used. In either case a toolchain used to build user-space applications must be selected, see the table below for a number of prepared configurations.

Table 3.1. Prepared Buildroot Toolchain Configurations

Config	Builds Toolchain	Libc	Target	Toolchain Location
<code>sfleonv8_shared_basic</code>	YES	uClibc	soft-float/v8	<code>dist/buildroot/build-br/host/usr/sparc-buildroot-linux-uclibc</code>
<code>hfleonv8_shared_basic</code>	YES	uClibc	hard-float/v8	<code>dist/buildroot/build-br/host/usr/sparc-buildroot-linux-uclibc</code>
<code>sfleonv8_multilib_glibc_basic</code>	NO	GLIBC	soft-float/v8	<code>/opt/sparc-linux-x.y.z-toolchains/multilib</code>
<code>hfleonv8_multilib_glibc_basic</code>	NO	GLIBC	hard-float/v8	<code>/opt/sparc-linux-x.y.z-toolchains/multilib</code>

3.3.2. Network Configuration

After building the Buildroot file system the first time the file system content is located in the Buildroot build directory `build-br/target`. Adding network settings for the network interfaces can be done by editing the `/etc/network/interfaces` file, for example setting `eth0` in DHCP and `eth1` to a static IP address is done by editing the interfaces file as follows:

```
# Configure Loopback
auto lo
```

```
iface lo inet loopback

# Do DHCP for ETH0
auto eth0
iface eth0 inet dhcp

# Static IP for ETH1
auto eth1
iface eth1 inet static
    address 192.168.1.207
    network 192.168.1.0
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
```

3.4. Linux configuration

Here the the number of threads used for kernel compilation for Linux kernel compilation can be set and whether Linuxbuild should set up file system source for Linux or not (using Buildroot as distribution, y is the way to go). Lastly, inside here the Linux kernel configuration is included.

3.5. LEON Linux Loader configuration

This is where one can configure how mklinuximg should be called. See mklinuximg documentation for details.

3.6. MKPROM2 configuration

This is where one can configure if and how MKPROM2 should be used. If selected, the path needs to be specified, unless mkprom2 is in PATH. Hardware specific parameters needs to be set to get a functional PROM image. See MKPROM2 documentation for details.

4. Building

After installing Linux, configuring Linuxbuild and all the selected components, the build process is started by typing **make build** in a shell, or by doubleclicking the build command inside the GUI under Step 3.

The resulting images and file system images are found in the `output/` directory. An image loadable from GRMON/GRMON2 is found in `output/images/image.ram`. Corresponding symbols can be found in `output/images/image`. If MKPROM2 is configured and used a PROM image can be found in `output/images/image.prom`.

After large configuration changes (e.g. changes in toolchain choice, FPU vs no FPU) or to remove any previously included Buildroot packages, rebuilding from scratch might be needed. To do that either type **make build** in a shell, or doubleclick the clean command inside the GUI under Step 3.

If the build process fail it may be due to that a required tool is missing. Build logs for different stages can be found in the respective build directories (e.g. `linux/build-linux/build.log` or `dist/buildroot/build-br/build.log`) or in the output directory (e.g. `output/images/mklinuximg.log` or `output/images/mkprom.log`).

5. Upgrading

To upgrade Linuxbuild itself, or components like the Linux kernel or Buildroot the user must do that manually. To save time the `linux/linux-src` and `dist/buildroot/buildroot-src` trees can be moved from the old Linuxbuild directory to the new, also the current configuration can be saved from the xconfig GUI and loaded into the new Linuxbuild installation or manually by copying the corresponding `.config` files.

The `mklinuximg` component can be upgraded when Aeroflex Gaisler release a new updated package. That upgrade process can be started from the KConfig GUI (`make xconfig`).

6. Support

For Support, contact the Aeroflex Gaisler support team at support@gaisler.com.
