LINUXBUILD User's Manual

LEON Linux build environment manual

LINBUILD
Version 1.0.8
November 2014

Kungsgatan 12
413 11 Gothenburg
Sweden

tel +46 31 7758650
fax +46 31 421407
www.aeroflex.com/gaisler

LINUXBUILD User's Manual

Copyright © 2014 Aeroflex Gaisler AB

Table of Contents

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 a graphical configuration utility, configuration files and a set of small Makefile-scripts for building and configuring some of the Linux tools that Aeroflex Gaisler provides support for.

The LINUXBUILD utility provides 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 or together from LINUXBUILD. 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.

- GNU Toolchain - GCC, BINUTILS, GLIBC
- 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 custom configurations can also be created by the user. Predefined configurations can be found in the `gaisler/configs` directory.

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

1.2. Requirements

- Linux host machine
- SPARC/LEON Linux 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.

1.3. Download Location

Table 1.1. Download locations

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/

1.4. Mini Quickstart Guide

Quick guide from zero to default configuration linux RAM image:

1. Download and install Linux toolchain
2. Download and unpack LINUXBUILD
3. change directory into newly unpacked directory
4. Run make xconfig
5. Click “Install Linux”
6. Double click “Execute Linux installation of latest stable leon-linux”

7. Click yes on popup and wait for installation to finish in separate window
8. Check that installation did not end with error and press return to close window which restarts xconfig
9. Scroll down to the end and click “Step3: build”
10. Double click “Execute make build”
11. Click yes on popup and wait for build to finish in separate window
12. Check that build did not end with error and press return to close window which restarts xconfig
13. Close xconfig

If all went well the following files can be found in `output/images:`

- `image.ram` — RAM image (for loading and running or for feeding to bootloader)
- `image` — virtual address image with symbols (for debug)

1.5. References

RD0	MKPROM2 User's Manual http://www.gaisler.com/doc/mkprom2.pdf
RD1	MKLINUXIMG User's Manual http://www.gaisler.com/anonftp/linux/linux-2.6/doc/mklinuximg-2.x.y.pdf
RD2	Prebuilt Toolchain http://www.gaisler.com/anonftp/linux/linux-2.6/toolchains/sparc-linux-x.y.z/
RD3	LEON Linux User's Manual http://www.gaisler.com/anonftp/linux/linux-2.6/doc/linux-x.y.z.pdf

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.

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:

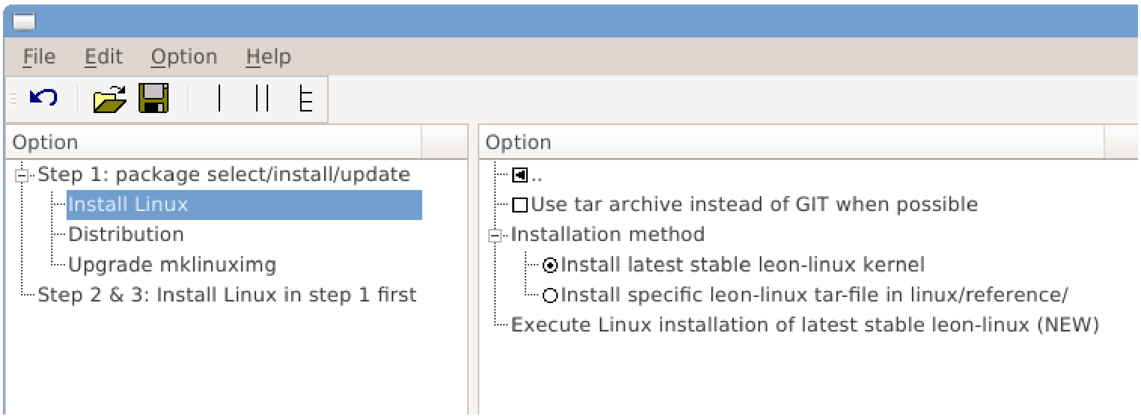
```
$ export PATH=/opt/sparc-linux-4.4.2-toolchains/multilib/bin:$PATH
$ which sparc-linux-gcc
/opt/sparc-linux-4.4.2-toolchains/multilib/bin/sparc-linux-gcc
```

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

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...
```



```
Step 1: package select/install/update
    Install Linux
    Distribution
    Upgrade mklinuximg
Step 2 & 3: Install Linux in step 1 first
```

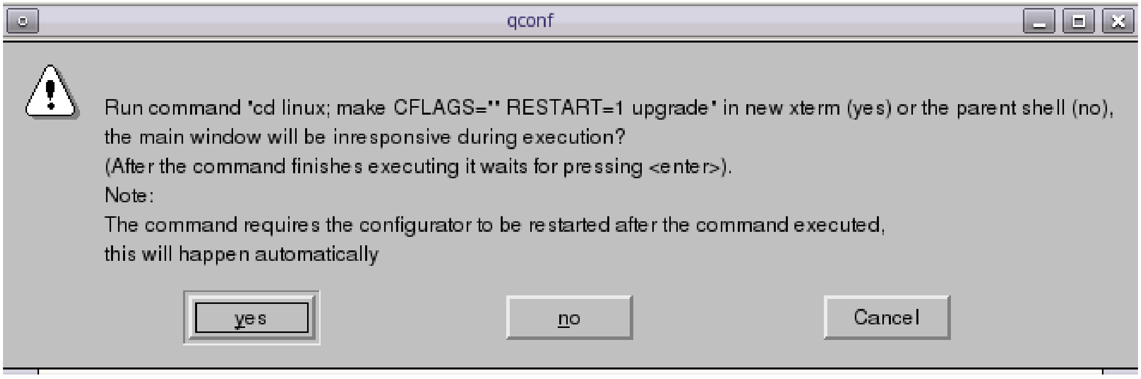
```
..
□ Use tar archive instead of GIT when possible
Installation method
    ● Install latest stable leon-linux kernel
    ○ Install specific leon-linux tar-file in linux/reference/
Execute Linux installation of latest stable leon-linux (NEW)
```

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 installation" a dialog will pop up asking whether to exeute the installation in a new xterm or in the parent shell:



Run command 'cd linux; make CFLAGS='' RESTART=1 upgrade' in new xterm (yes) or the parent shell (no), the main window will be inresponsive during execution?
(After the command finishes executing it waits for pressing <enter>).
Note:
The command requires the configurator to be restarted after the command executed, this will happen automatically

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

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

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.

This option can for example be useful when using custom scripts to generate a root file system, of the kernel uses a file system over NFS or located in FLASH/PROM.

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. MKPROM2

The mkprom utility is used to generate FLASH/PROM bootable images. It is downloaded from Aeroflex Gaisler website and installed manually. This is a separate package and can be used with other LEON tools and Operating Systems than LINUXBUILD. The MKPROM2 utility is found through the PATH environment variable or a custom absolute path set by the configuration.

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 listed below.

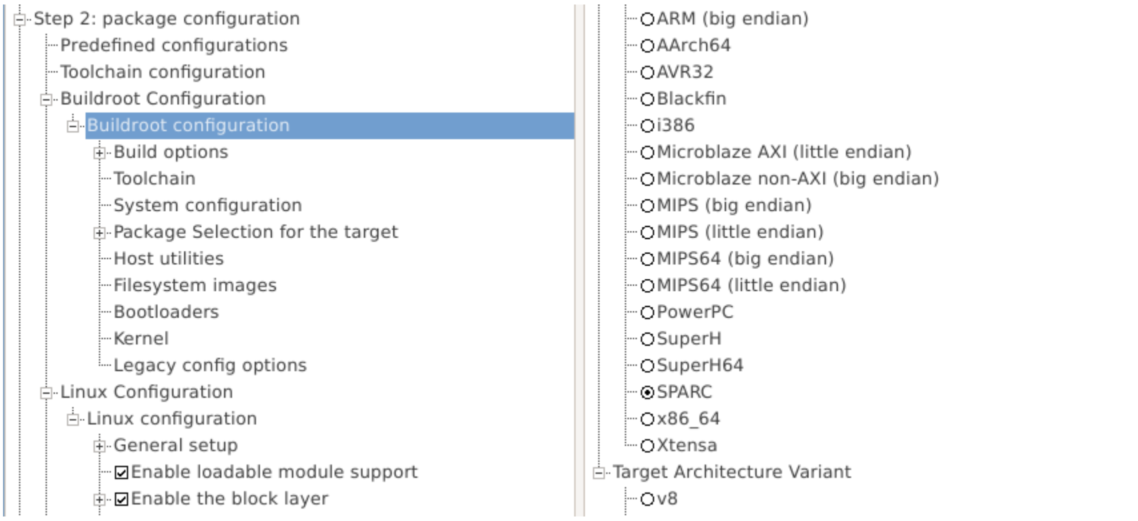
Using xconfig is the recommended configuration interface:

- **make xconfig** - Qt based GUI (Qt-3/4 libs required)

Not recommended, but provided as is:

- **make gconfig** - GTK based GUI
- **make menuconfig** - ncurses based terminal interface

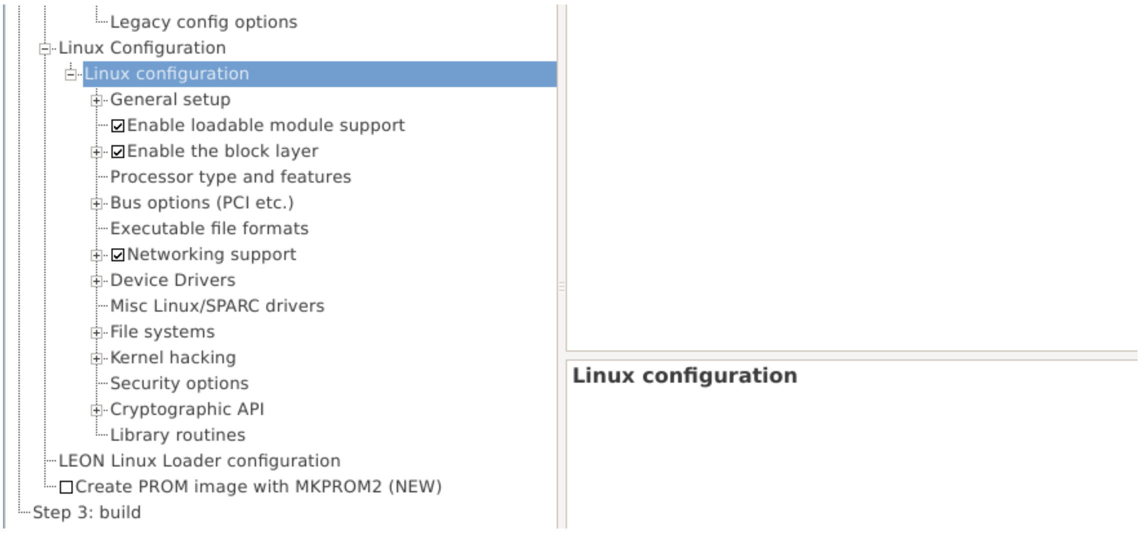
The buildroot and Linux configuration trees are displayed as subtrees inside the main screen:



```
⊟ Step 2: package configuration
    Predefined configurations
    Toolchain configuration
  ⊟ Buildroot Configuration
    ⊟ Buildroot configuration
      ⊞ Build options
        Toolchain
        System configuration
      ⊞ Package Selection for the target
        Host utilities
        Filesystem images
        Bootloaders
        Kernel
        Legacy config options
  ⊟ Linux Configuration
    ⊟ Linux configuration
      ⊞ General setup
        ☑ Enable loadable module support
      ⊞ ☑ Enable the block layer

    ○ ARM (big endian)
    ○ AArch64
    ○ AVR32
    ○ Blackfin
    ○ i386
    ○ Microblaze AXI (little endian)
    ○ Microblaze non-AXI (big endian)
    ○ MIPS (big endian)
    ○ MIPS (little endian)
    ○ MIPS64 (big endian)
    ○ MIPS64 (little endian)
    ○ PowerPC
    ○ SuperH
    ○ SuperH64
    ● SPARC
    ○ x86_64
    ○ Xtensa
⊟ Target Architecture Variant
    ○ v8
```

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



```
        Legacy config options
  ⊟ Linux Configuration
    ⊟ Linux configuration
      ⊞ General setup
        ☑ Enable loadable module support
      ⊞ ☑ Enable the block layer
        Processor type and features
      ⊞ Bus options (PCI etc.)
        Executable file formats
      ⊞ ☑ Networking support
      ⊞ Device Drivers
        Misc Linux/SPARC drivers
      ⊞ File systems
      ⊞ Kernel hacking
        Security options
      ⊞ Cryptographic API
        Library routines
    LEON Linux Loader configuration
    ☐ Create PROM image with MKPROM2 (NEW)
  Step 3: build

Linux configuration
```

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 cofiguration 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 cofiguration 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).

Note that SPARC V7 systems are not supported by Linux kernel or Buildroot.

3.3.1. Buildroot toolchain configuration

By default, en 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
sfleonv8_shared_basic	YES	uClibc	soft-float/v8	`dist/buildroot/build-br/host/usr/sparc-buildroot-linux-uclibc`
hfleonv8_shared_basic	YES	uClibc	hard-float/v8	`dist/buildroot/build-br/host/usr/sparc-buildroot-linux-uclibc`
sfleonv8_multilib_glibc_basic	NO	GLIBC	soft-float/v8	`/opt/sparc-linux-x.y.z-toolchains/multilib`
hfleonv8_multilib_glibc_basic	NO	GLIBC	hard-float/v8	`/opt/sparc-linux-x.y.z-toolchains/multilib`

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.

LEON specific Linux kernel configuration options are documented separatley in the LEON Linux User's Manual [RD3].

3.5. LEON Linux Loader configuration

This is where one can configure how mklinuximg should be called. See mklinuximg documentation for details [RD1].

This section linkes the Linux kernel to main memory and sets up the LEON Linux RAM loader responsible for setting up a low-level environment required for the SPARC Linux kernel.

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 [RD0].

The resulting PROM image can be loaded to FLASH using the GRMON flash commands **flash unlock**, **flash erase** and **flash load**. The PROM image can also be loaded into TSIM or GRSIM and executed.

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 double clicking the build command inside the GUI under Step 3.

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

After major configuration changes (e.g. changes in toolchain choice, FPU vs no FPU) or removal of any previously included Buildroot packages, rebuilding from scratch might be required. To do that either type **make clean** in a shell, or double click the clean command inside the GUI under Step 3.

If the build process fails 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.

7. Disclaimer

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