



# ***Introduction to uClinux*** **Training lab book**

*Thomas Petazzoni, Michael Opdenacker*  
*Free Electrons*  
<http://free-electrons.com>





## About this document

This document is part of an embedded Linux training from Free Electrons.

You will find the whole training materials (slides and lab book) on <http://free-electrons.com/training>

Lab data can be found on [http://free-electrons.com/labs/embedded\\_linux.tar.bz2](http://free-electrons.com/labs/embedded_linux.tar.bz2).

## Copying this document

© 2009, Free Electrons, <http://free-electrons.com>.



This document is released under the terms of the [Creative Commons Attribution-ShareAlike 3.0 license](https://creativecommons.org/licenses/by-sa/3.0/). This means you are free to download, distribute and even modify it, under certain conditions.

Document updates and translations available on [.http://free-electrons.com/docs/uclinux](http://free-electrons.com/docs/uclinux).

Corrections, suggestions, contributions and translations are welcome!

## Training setup

See the training labs on <http://free-electrons.com/training/drivers> for setup instructions, which are shared with these practical labs.



## uClinux / lab1 - Using uClinux-dist

Objective: use the uClinux-dist distribution to compile a kernel and a root filesystem for a virtual MMU-less ARM system

After this lab, you will be able to

- Install a cross-compiling toolchain for uClinux
- Configure the uClinux-dist distribution
- Run the uClinux-dist build tool to generate a kernel image and root filesystem for a virtual MMU-less ARM system.
- Boot your virtual system with the kernel and root filesystem that you created.

### Lab setup

Go to the `/mnt/labs/uclinux/lab1` directory.

In `/mnt/labs/uclinux/lab1/`, you will find:

- Useful stuff...

### Target system

We are going to use the SkyEye (<http://skyeye.org>) emulator to run a virtual AT91 ARM7TDMI EB01 board. SkyEye's engine is derived from the GNU Debugger (GDB), and was once called GDB/Armulator.

Install the SkyEye emulator:

```
apt-get install skyeye
```

### uClinux toolchain

Download the `arm-linux-tools-20070808` toolchain from <http://ftp.snapgear.org/pub/snapgear/tools/arm-linux/>.

This is a big file (272 MB). In case the SnapGear website is down or slow, you can also download this toolchain from <http://free-electrons.com/labs/tools>, or directly get it from your instructor on a USB flash disk.

With root permissions, extract this toolchain in the `/` directory.

You will see that this toolchain installs directly in `/usr/local`, which is a standard location. Therefore, you won't even have to set a special `PATH` configuration.

### uClinux-dist setup

First, download `uclinux-dist-20080808` from <http://uclinux.org/pub/uClinux/dist/>, from the uClinux SourceForge mirror or from <http://free-electrons.com/labs/tools>. Since this is also a big file (282 MB compressed with `bzip2`), you may also directly get it from your instructor.

Extract the archive in your lab directory (`/mnt/labs/uclinux/lab1`).

Also install the `genromfs` tool, which will be useful to generate a

Toolchains usually install in custom directories. This makes it possible to use different toolchains. Here, installing the toolchain in `/usr/local` could prevent from using other toolchains with the same filenames, in the same location.



romfs filesystem image.

```
apt-get install genromfs
```

## uClinux-dist configuration

Go to the `uClinux-dist` directory.

Start the configuration interface, by running `make xconfig` or `make menuconfig`.

In `Vendor/Product Selection`, choose GDB as the vendor. Then, in `GDB Products`, choose SkyEye.

In `Kernel/Library/Defaults Selection`, enable `Customize Kernel Settings` and `Customize Application/Library Settings`.

Then, quit the interface (`File -> Quit`), accepting to save your configuration. After a few seconds, the `uClinux-dist` Makefile will then run the configuration interfaces for the kernel and for applications and libraries.

In the kernel configuration interface, set the kernel command line (`CONFIG_CMDLINE` in the `Boot options` menu) to `init=/bin/sh`. Save your settings and quit this interface.

Now, the applications and library configuration interface appears. Have a look at the default settings. These settings will generate a root filesystem that works and is sufficient for the needs of this lab.

## Building the root filesystem

`uClinux-dist` makes it very easy to do! You just need to run:

```
make
```

After a few minutes (5 to 15, according to how fast your workstation is), the build job is complete.

Look at the `images/` directory. In our case, you will find:

- `romfs-inst.log`  
A report of the creation of the filesystem. Shows the files inside.
- `linux`  
The image of the Linux for your target.
- `boot.rom`  
The root filesystem image (a romfs filesystem in our case).

Of course, the contents of the `images/` directory depends on the target you chose.

## Booting your system

Go back to the main lab directory (`/mnt/labs/uClinux/lab1`).

Have a look at the `skyeeye-at91.conf` file. It describes how the AT91 EB01 board is emulated. You can see that parts of its address space maps to the contents of the `./uClinux-dist/images/boot.rom` file. This corresponds to the ROM of the virtual board.

You can try to start your virtual machine now:

```
skyeeye -e uClinux-dist/images/linux -c skyeeye-at91.conf
```

Congratulations if your board booted successfully!

Unlike QEMU, SkyEye can't pass parameters to a Linux kernel. Hence, you have to hardcode Linux kernel parameters at kernel compile time.

SkyEye options:  
-e: (ELF executable format) kernel file  
-c: SkyEye configuration file.



### Going further

Here are other things you could do if you finished before the others

- Try to add other commands to the root filesystem. In particular, you compile BusyBox commands.
- Uncomment the network related commands in the SkyEye configuration file, and try to ping the host from the virtual target.