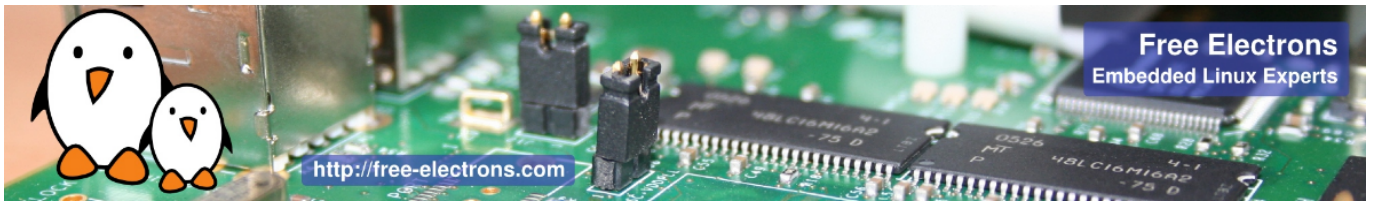


# Embedded Linux development with Buildroot training

## 3-day session

<b>Title</b>	<b>Embedded Linux development with Buildroot training</b>
<b>Overview</b>	<ul style="list-style-type: none"> <li>Introduction to Buildroot</li> <li>Managing and building the configuration</li> <li>Buildroot source and build trees</li> <li>Toolchains in Buildroot</li> <li>Managing the Linux kernel configuration</li> <li>Root filesystem</li> <li>Download infrastructure</li> <li>GNU Make 101</li> <li>Integrating new packages</li> <li>Advanced package aspects</li> <li>Analyzing the build</li> <li>Advanced topics</li> <li>Application development with Buildroot</li> <li>Understanding the Buildroot internals</li> <li>Buildroot community: support and contribution</li> <li>What's new in Buildroot ?</li> </ul>
<b>Duration</b>	<b>Three</b> days - 24 hours (8 hours per day). 40% of lectures, 60% of practical labs.
<b>Trainer</b>	<b>Thomas Petazzoni.</b> Thomas is a major Buildroot developer since 2009, with more than 2700 patches integrated and an active participation to the development process.
<b>Language</b>	Oral lectures: English, French. Materials: English.
<b>Audience</b>	Companies already using or interested in using Buildroot to build their embedded Linux systems.
<b>Prerequisites</b>	<p><b>Knowledge of embedded Linux</b> as covered in our embedded Linux course:  <a href="http://free-electrons.com/training/embedded-linux/">http://free-electrons.com/training/embedded-linux/</a></p> <p><b>Knowledge and practice of Unix or GNU/Linux commands</b>            People lacking experience on this topic should get trained by themselves, for example with our freely available on-line slides:  <a href="http://free-electrons.com/blog/command-line/">http://free-electrons.com/blog/command-line/</a></p>

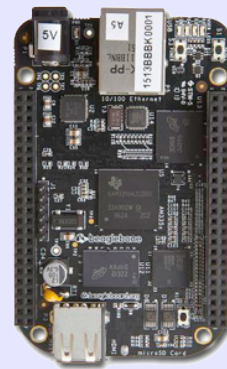


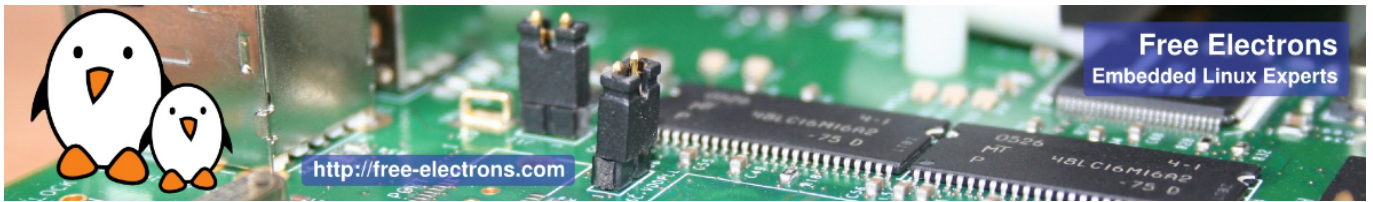
<b>Required equipment</b>	<p><b>For on-site sessions only.</b> Everything is supplied by Free Electrons in public sessions.</p> <ul style="list-style-type: none"><li>• Video projector</li><li>• PC computers with at least 4 GB of RAM, and Ubuntu Linux installed in a <b>free partition of at least 30 GB. Using Linux in a virtual machine is not supported</b>, because of issues connecting to real hardware.</li><li>• We need Ubuntu Desktop 14.04 (32 or 64 bit, Xubuntu and Kubuntu variants are fine). We don't support other distributions, because we can't test all possible package versions.</li><li>• <b>Connection to the Internet</b> (direct or through the company proxy).</li><li>• <b>PC computers with valuable data must be backed up</b> before being used in our sessions. Some people have already made mistakes during our sessions and damaged work data.</li></ul>
<b>Materials</b>	Print and electronic copies of presentations and labs. Electronic copy of lab files.

## Hardware

The hardware platform used for the practical labs of this training session is the **BeagleBone Black**, which features:

- An ARM AM335x processor from Texas Instruments (Cortex-A8 based), 3D acceleration, etc.
- 512 MB of RAM
- 2 GB of on-board eMMC storage (4 GB in Rev C)
- USB host and device
- HDMI output
- 2 x 46 pins headers, to access UARTs, SPI buses, I2C buses and more.





## Day 1 - Morning

---

### Lecture - Embedded Linux and build system introduction

- The general architecture of an embedded Linux system
- Build systems vs. binary distributions
- Role of a build system
- Comparison of existing build systems

### Lecture - Introduction to Buildroot

- Key facts about the project
- Getting Buildroot
- Basic configuration of Buildroot
- Doing a first build

### Lab - Basic Buildroot usage

- Getting and setting up Buildroot
- Configuring and building a basic system with Buildroot for the BeagleBone Black
- Flash and test the generated system on the BeagleBone Black

### Lecture - Managing the build and configuration

- Out of tree build
- Using and creating *defconfigs*
- Defconfig fragments
- Other building tips

## Day 1 - Afternoon

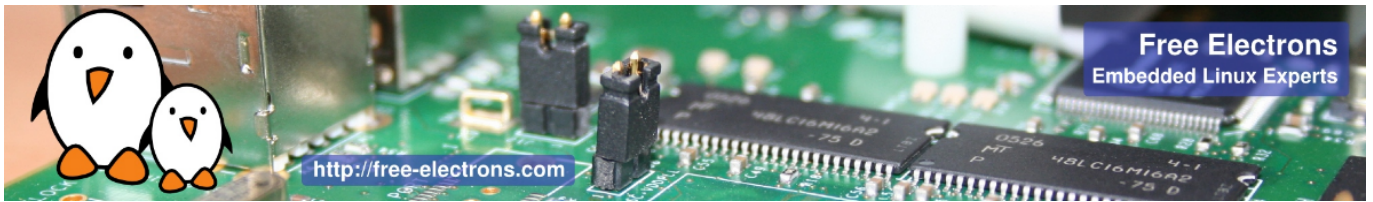
---

### Lecture - Buildroot source and build trees

- Details about the Buildroot source code organization
- Details about the Buildroot build tree

### Lecture - Toolchains in Buildroot

- The different choices for using toolchains in Buildroot
- Overview of the toolchain options
- Using existing binary toolchains, such as Sourcery CodeBench toolchains, understanding *multilib* capabilities and integration of toolchains in Buildroot
- Generating custom toolchains with *Crosstool-NG*, and re-use them as external toolchains



## Lecture - Managing the Linux kernel configuration

- Loading, changing and saving the kernel configuration

## Lecture - Root filesystem construction in Buildroot

- Understand how Buildroot builds the root filesystem: *skeleton*, installation of packages, overlays, *post-build* and *post-image* scripts.
- Customization of the root filesystem contents
- System configuration: *console* selection, various */dev* management methods, the different *init* implementations, etc.
- Understand how Buildroot generates filesystem images

## Lab - Root filesystem customization

- Explore the build output
- Customize the root filesystem using a *rootfs overlay*
- Customize the kernel with patches and additional configuration options
- Add more packages
- Use *defconfig* files and *out of tree* build

## Day 2 - Morning

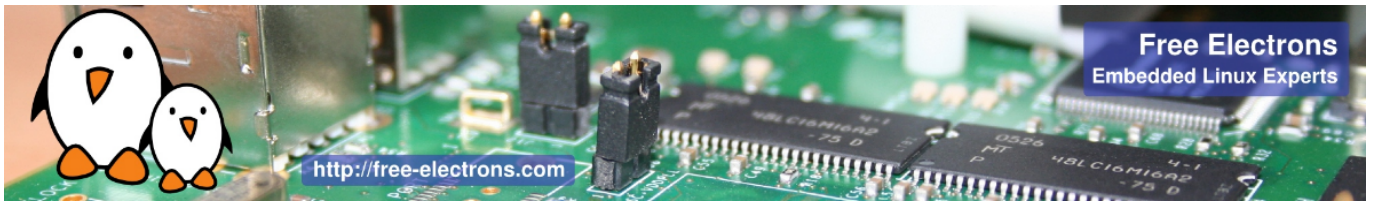
---

### Lecture - Download infrastructure in Buildroot

- Downloading logic
- Primary site and backup site, doing offline builds
- VCS download, integrity checking
- Download-related *make* targets

### Lecture - GNU Make 101

- Basics of make rules
- Defining and referencing variables
- Conditions, functions
- Writing recipes



## Lecture - Integrating new packages in Buildroot

## Lab - New packages in Buildroot

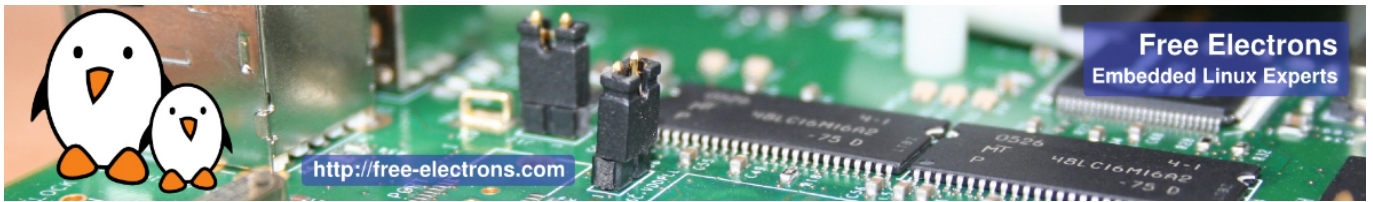
- How to integrate new packages in the Buildroot configuration system
  - Understand the different package infrastructures: for *generic*, *autotools*, *CMake*, *Python* packages and more.
  - Writing a package `Config.in` file: how to express dependencies on other packages, on toolchain options, etc.
  - Details on writing a package recipe: describing the package source code location, download method, configuration, build and installation steps, handling dependencies, etc.
- Create a new package for *nInvaders*
  - Understand how to add dependencies
  - Add patches to *nInvaders* for *Nunchuk* support

## Day 2 - Afternoon

### Lecture - Advanced package aspects

### Lab - Advanced packages

- Licensing report
  - Patching support: patch ordering and format, global patch directory, etc.
  - User, permission, device tables
  - Init scripts and systemd unit files
  - Config scripts
  - Understanding *hooks*
  - Overriding commands
  - Legacy handling
  - Virtual packages
- Package an application with a mandatory dependency and an optional dependency
  - Package a library, hosted on GitHub
  - Use *hooks* to tweak packages
  - Add a patch to a package



## Day 3 - Morning

---

### Lecture - Analyzing the build: licensing, dependencies, build time

- Usage of the legal information infrastructure
- Graphing dependencies of packages
- Collecting and graphing build time information

### Lecture - Advanced topics

- BR2\_EXTERNAL to store customizations outside of the Buildroot sources
- Package-specific targets
- Understanding rebuilds
- Tips for building faster

### Lab - Advanced aspects

- Use build time graphing capabilities
- Use dependency graphing capabilities
- Use licensing report generation, and add licensing information to your own packages
- Use BR2\_EXTERNAL

## Day 3 - Afternoon

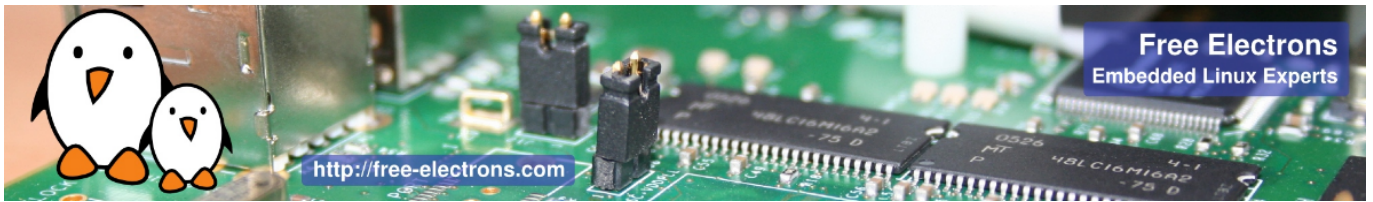
---

### Lecture - Application development with Buildroot

- Using Buildroot during application development
- Usage of the Buildroot environment to build applications outside of Buildroot
- Generate an SDK for other developers
- Remote debugging with Buildroot

### Lab - Application development with Buildroot

- Build and run your own application
- Remote debug your application
- Use `<pkg>_OVERRIDE_SRCDIR`
- Set up Eclipse for Buildroot application development



## Lecture - Understanding Buildroot internals

- Detailed description of the Buildroot build process: toolchain, packages, root filesystem construction, stamp files, etc.
- Understanding virtual packages.

## Lecture - Getting support and contributing, what's new in Buildroot

- Getting support: *Bugzilla*, *mailing list*, *IRC*
- Contributing: understanding the development process, how to submit patches
- What's new in Buildroot: summary of the major changes since the last two years