

Hands-on course , 4
day(s)
Ref : LXT

Participants

Every step of the training session is immediately applied as a case study on an embedded ARM board with a touch screen to test graphical developments.

Pre-requisites

Participants should be familiar with UNIX/Linux command line. The knowledge of the C language is optional.

Next sessions

Embedded and real-time Linux

OBJECTIVES

This training targets engineers who wish to use Linux in embedded systems with industrial realtime issues. During the session you will learn to build your own embedded system from boot-loader and Linux kernel up to application cross-development using free open-source software.

1) Open-source cross development tools

2) The universal Boot loader: uBoot

3) Linux kernel

4) Root File system

5) Linux and Real-Time

Workshop

Every step of the training session is immediately applied as a case study on an embedded ARM board with a touch screen to test graphical developments.

1) Open-source cross development tools

- Architecture overview. Overview of an embedded system and of the Linux kernel architecture.
- Cross development tool chain, gcc cross compiler, C libraries, glibc and uClibc, GNU debugger, GNU ELF tools.
- Embedded development tools, QEMU, BUILDROOT, BUSYBOX.

Workshop

Installation and practice of Buildroot to build a cross compilation toolchain.

2) The universal Boot loader: uBoot

- uBoot project overview. A walk through the source code. Supported architectures.
- Basic functionalities. The ulmage format for booting uBoot Images.
- Practice on a ARM-9 device with uBoot. Configuration, compilation and installation in a qemu sandbox for testing.
- The uBoot configuration options. uBoot commands and environment.
- Integration of new uBoot commands. Development of a standalone program using uBoot as BIOS.

Workshop

You will add a new command to uBoot and test uBoot inside QEMU.

3) Linux kernel

- Licenses implications on kernel and kernel modules development.
- Kernel development tools, quilt, GDB, GIT, LTT. Kernel configuration tool Kbuild.
- Kernel configuration and compilation. Module development and compilation.
- The Linux boot process. The schedulers, preemption and scheduling.
- The Linux driver framework and standard drivers.
- VFS Essentials callbacks, implementing a quick and dirty driver with open, read and write callbacks.

Workshop

How to modify the kernel tree to add a new driver to the kernel tree. Generate a patch formatted for the LKML. Develop a character driver outside of the kernel tree.

4) Root File system

- Bottom-up approach for the root file system creation.
- Adding users management with busybox.
- Application integration.
- Copying the root file system on a memory technology device.

Workshop

Create rootfs from scratch using busybox and test it on a real ARM target. Use buildroot to build new generic applications. Add your own application.

5) Linux and Real-Time

- History and kernel evolution.
- Linux-RT a realtime evolution.
- Nano kernels and interrupt virtualisation.
- Xenomai applications development.
- Real-time Hypervisors.

Workshop

Installation of Xenomai, development of a realtime application with Posix and Native Interface. Benchmark comparing Linux standard applications and Xenomai applications under heavy load.