

Introduction

Getting started with embedded Linux development can be a time consuming process. This course lasting two days will kick-start you by showcasing all the tools and processes. The goal of the course is to show all the steps involved in building an embedded Linux project and to prepare you for further studies of the topic. All the labs are performed on an emulated embedded platform based on an Arm CPU.

By the end of the course you will know:

- How to setup an embedded Linux development environment.
- How to acquire, configure and build a Linux kernel.
- How to extend the Linux kernel with a custom driver.
- How to create a basic root file system based on busybox.
- How to setup the u-boot loader to boot your kernel with your root file system.
- How to use an embedded Linux toolkit, encapsulating the process into one tool.
- The legal aspects of using open source software.

Prerequisites:

- Basic knowledge of Linux (as a user).
- Basic knowledge of embedded systems.
- Basic knowledge of the C programming language.

Schedule

The training session spans two days and contains four labs. Each day is split into four sessions with short breaks in-between. If you order an on-site training, you can choose which section(s) to put extra focus on. We can also provide a third day of training for a specific hardware platforms upon request.

Day 1

Session 1 — Introduction to Embedded Linux

- Introduction to this course
- Background to embedded Linux
- Desktop Linux vs. embedded Linux
- A typical embedded Linux system
- Working in an open source environment

Session 2 — The development environment

- The components of an embedded Linux development environment
- Choosing a C library
- Building an environment
- Available tools for automated builds

Lab 1 — Building a cross compilation tool-chain

- Using crosstool-ng to build a GCC toolchain

Session 3 — The architecture of the Linux kernel

- Introduction to the kernel
- The scheduler
- File systems
- Device drivers
- Real-time aspects

Session 4 — Configuration and building

- Using existing configurations
- Pros and cons of modules
- The right device drivers
- Cross compiling the kernel

Lab 2 — Configuring and building a basic Linux system

- Setting up the build environment
- Configuring and building the kernel
- Running the kernel in an emulator

Day 2

Session 5 — Creating a user-space

- The parts of user-space
- BusyBox
- Custom applications
- Configuring init
- Device nodes and mount points
- More applications in brief

Session 6 — Adding custom drivers

- The anatomy of a Linux driver
- Registering I/O and memory regions
- Communicating with user-space applications
- Building and loading the driver
- Legal aspects

Lab 3 — Building a custom Linux system

- Building a custom device driver
- Building a custom application
- Building a custom root file system
- Building a kernel for the system

Session 7 — The boot loader

- The boot process
- A look at available boot loaders
- Features of u-boot
- Configuring and building u-boot
- Using u-boot to load a Linux kernel

Session 8 — A look at frameworks for embedded Linux

- Distributions and frameworks
- Available frameworks
- The structure of Buildroot
- Configuring a Buildroot environment

Lab 4- Building an embedded system using Buildroot

- Adding your own code to your Buildroot project
- Configuring and building