

Workshop

Compact Embedded Linux

In the development of processor based embedded systems Linux is often the operating system of choice. But how can I design my custom embedded Linux system for a specific hardware platform like the XILINX ZYNQ SoC? All aspects like development tools chains, boot loader, root file system and Linux kernel for embedded systems are discussed in this seminar. Those components will be configured, compiled and used by the participants. Only free tools like the GNU compiler, u-boot boot loader, or BusyBox are used. The participants learn how to create and adapt the components of an embedded Linux system, so that they are able to create a Linux system tailored for their specific applications.

This course is practical orientated so that each section will end with an exercise to establish a deeper understanding of the training material. The target platform for this training is the XILINX ZYNQ, but the learned tools and

techniques can be applied to other technologies such as processor systems as well.

Applicable technologies

Processor based embedded Linux systems like XILINX ZYNQ and others.

Requirements

A basic understanding of using Linux as an operating systems is beneficial, but not required. Linux basics with a focus on Hardware developers is discussed in this training. Participants should know basics about Makefiles and the C-Programming language.

Duration and Costs:

Duration: 3 days

Cost: € 2.100,- net per person, including detailed training material, drinks in the breaks and lunch

Agenda

Linux Basics for Hardware developers

- Linux Root filesystem concept
- Using the Console
- Scheduler, Processes and tasks
- Cross compiler and libraries
- Patching

Boot loader

- General boot process
- U-boot configuration and compilation

- Hardware debugging with u-boot
- Network boot for practical development

Linux kernel and root file system

- Linux kernel architectural basics
- Boot process
- Flatten Device Tree support
- Linux folder hierarchy
- File systems and wear leveling
- C library glibc, eglibc, uclibc