As of 09.02.2026

Embedded Real-Time Linux from Bootloader to Real-Time System Using Yocto (Linux RTOS) - Face-to-Face Training

Objectives

If you have to build an embedded Linux target - how do you begin? What do you need? How do you get a real-time system?

This embedded Linux training focuses on the setup and operation of an embedded Linux system with hard real-time requirements.

We start with the setup and function of the Yocto build system, generating an own layer with an own machine, image and distro. This provides a foundation for an own board support package (BSP).

Afterwards, the components of an embedded Linux system are highlighted in detail, from the ROM loader to the bootloader, the Linux kernel with device tree to the root filesystem.

Like in everyday project work, configurations are adjusted, patches created and extensions implemented. All this happens in the Yocto layer implemented before, so that the entirely adjusted system is available in the Yocto layer for further use.

Based on the system created, requirements like boot time optimization and over-the-air updates are discussed and explained.

The tools used are available for a wide range of architectures.

Only freely available open source software is used in the seminar.

Participants

Software developers and hardware developers

Requirements

Profound ANSI-C programming knowledge; good basic knowledge of Linux is of advantage.

Embedded Real-Time Linux from Bootloader to Real-Time System Using Yocto (Linux RTOS) - Face-to-Face Training

Content

Development Environment

- Cross-development toolchain
- Creating a toolchain, bootloader, kernel and root filesystem
- Comparison of build systems: Yocto, buildroot, Debian-based

Yocto as Build System

- Setup of Yocto
- bitbake as build program
- Recipes, classes and configurations
- Building own layers
- Generating BSP with machine
- Target system with image and software selection with distro
- Analysis of the build process, log and run files
- Diagnosis of build problems

- wic - Open Embedded Image Creator

Bootloader

- Linux boot process from the ROM loader to the login prompt
- U-boot and barebox as bootloader
- Configuration of the bootloader
- Creation of patches and their integration in Yocto

Device Tree

- Hardware description in the device tree
- Syntax and application
- GPIO controller
- Pin multiplexing
- I2C and SPI bus
- Extensions with own devices

Linux Kernel

- Kernel configuration
- Adjustments to specific boards and projects
- Application of kernel drivers
- Creation of patches
- Integration of adaptations in Yocto
- Hard real-time with Linux (PREEMPT_RT)
- Threaded interrupts and scheduling

Root Filesystem

- Init daemons: systemd, system-V, busybox init
- C libraries: glibc, uClibc
- Creation and integration of own C programs
- Configuring daemons for the target and specific Yocto recipes

System Design

- Minimum systems with busybox
- RAM disk, initial RAM filesystem and initrd
- NAND and NOR with MTD drivers
- Flash filesystem (UBIFS, JFFS2)
- Managed (FTL) flash with ext4
- Read-only with squashfs and writing in overlayfs
- Reproducible build process
- Measuring and optimizing boot times
- System update over-the-air (OTA), configuration in Yocto

Exercises

- All exercises are performed on an ARM Cortex-A8 (AM-335x) using freely accessible open source tools.
- Practical exercises on all topics enable you to directly apply what you just learned
- After the seminar, you have built an entire embedded real-time Linux with Yocto.

FACE-TO-FACE TRAINING

Price *	Duration
3.300,00 €	5 days

Training code: E-LIN-RTD
* Price per attendee, in Euro plus VAT

Face-To-Face - German

Date	Duration
08.06. – 12.06.2026	5 days

Coaching

Our coaching services offer a major advantage: our specialists introduce their expertise and experience directly in your solution process, thus contributing to the success of your projects.

As of 09.02.2026

We will be happy to provide you with further information or submit a quotation tailored to your requirements.