

# **Embedded Linux Software Development - Live Online Training**

## Objectives

This training focuses on the development of software for embedded Linux.

It highlights all aspects of system-near development and offers exercises using the related development tools.

Attendees get to know the Posix operating system and can handle the key development and diagnostic tools. The exercises comprise the operating system mechanisms as well as development and diagnostic tools.

#### Participants

Software developers, software architects

#### Requirements

Profound ANSI-C programming knowledge as well as good basic knowledge of Linux. Good C programming knowhow as well as proficiency in using the Linux Shell (e.g. ls, cp, mv, dd) with input/ output redirection.

# Live-Online-Training

\* Price per attendee, in Euro plus VAT

Training code: LE-LIN-SWE

## Face-To-Face - English

Duration 4 days

## Live Online - German

**Duration** 4 days

## Face-To-Face - German

**Date Duration** 09.12. – 12.12.20244 days 31.03. – 03.04.20254 days

# **Embedded Linux Software Development - Live Online Training**

## Content

## System-Near Software Development

- Files, pipes and device nodes
- Processes, CPU affinity
- Scheduling; RT, deadline, batch task
- Processes, signals, core dump
- Shared memory, memory mapping

© MicroConsult Microelectronics Consulting & Training GmbH More trainings on www.microconsult.com. Subject to change. All prices per attendee, in EUR plus VAT. Contact: info@microconsult.com, phone +49 (0)89 450617-71



As of 20.05.2024

- Semaphore, message queue
- Multithreading
- Mutex, robust mutex, PI mutex, RW lock, barrier
- Hrtimer framework and Posix timer
- Hardware interfaces: GPIOs, I2C

#### **Development Environment and Diagnostic Tools**

- Cross development toolchain
- Cross debugging with gdb and gdbserver
- proc, sys and debug FS
- Memory leaks, memory overwrite; valgrind
- Code coverage analysis and profiling; gcov and gprof
- Ptrace interface of the Linux kernel; debugger operation
- strace and Itrace operation and use
- Operation of the function trace frameworks (ftrace)
- Tracing of interrupt and scheduling events

## Hardware

- All exercises are performed on a phyBOARD with ARM Cortex-A8 (AM-335x) using freely accessible open source tools (remote access).