

Embedded Linux in Theory and Practice



Training Description

The start into a powerful tool as Linux is important. The aim of the training is to provide you basic concepts in embedded Linux, as well as the handling of Linux.

You will learn the advantages and disadvantages of Linux and which components are necessary to build an embedded GNU / Linux system. Furthermore at the end of the training you will know, where you can get those components from and how to configure, translate and install them. You will also learn how to get support in the case of uncertainty and which license models are used in the Linux environment.

By means of numerous hands-on examples you will learn how to build an embedded GNU / Linux System out of the mainline components.

Target Group

Project manager, software, hardware and system engineers

Prerequisites

Basic knowledge of how to use Linux (Ubuntu), familiarity with embedded C concepts and programming, the ability to develop software using C-Syntax, also basic knowledge of embedded hardware (Eva Boards) is of advantage.

Training Content

1st day: basics

- Introduction into GNU/ Linux: History, licenses, standards, working with open source, spelunking, Unix, philosophy
- Characteristics of Embedded Linux:
 - Embedded Systems
 - Comparison of Embedded Linux vs. Desktop Linux
 - Identifying and solving dysfunctionalities
 - Portability
 - Building for the target: tool-chains, C-Libraries

- Eval Board - Beagle Bone Black
 - Booten (generic) or rather the Beagle Bone Black
 - Partitioning or rather formatting of SD cards
 - Partition with boot-loader, kernel, filling of rootfs
 - Configuring serial consoles
 - Booting a board with Linux/GNU

2nd day: installing and configuring the Host and the Target

- Installing of the toolkit, the NFS server, tftp server
- U boot: check out, configuration, cross compiling, installation
- Flattened device tree
- GNU/Linux kernel: ulmage, check out, configuration, cross compiling, installation, kernel modules
- Root File System
- Adjustments: Adding network support to the board (U boot scripting, network support in U boot, customized kernel with network support)
- Rootfs via NFS
- Init (Sys-V, Upstart, Initng, Systemd), Bootgraph, Bootchart

3rd day: kernel modules, an overview of drivers and debugging

- Kernel module: Hello Kernel, module-init-tools, Kconfig, Kbuild, out of tree, in tree
- Device driver: Device nodes, writing a character driver, registration, initialization, Miscellaneous Character driver
- An overview of debugging / profiling / tracing:
 - Simple debugging tools: lsof, ltrace, strace, proc, top, netstat, syslog
 - Further debugging tools: gdb and target gdb, gdbserver, kgdb/kdb and agent-proxy, JTAG
 - Profiling: time, gprof, gcov, oprofile
 - Tracing: race, kernelshark, LTTng
 - Multiple other tools: top, latencytop, powertop, powerdebug, crash, systemtap

Method and Training Materials

Presentation and practical examples with host (Laptops with Ubuntu 14.04.x LTS) and target system (e.g. Beagle Bone Black Rev. C - <http://beagleboard.org/BLACK>)

The electronic devices will be provided during the training. There will be one workspace for two participants. You will receive a workbook (in English), a Beagle Bone Black Rev. C with standard FTDI 3.3V to USB cable. After the training you will be given a download link with examples for more in-depth study.

Duration and Training Fee

3 days, 1.790 Euro plus VAT tax per person

The fee includes the training certificate, training materials, snacks, beverages and lunch.

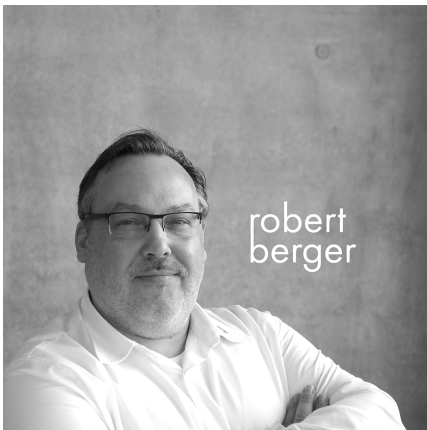
Number of Participants

6 to 12 participants

Location

Technology center TechBase, Franz-Mayer-Straße 1, D-93053 Regensburg

Trainer



Robert Berger

Since 1993, Robert Berger gathered experience in software design and development for embedded systems. He has used GNU/Linux on desktop and server class machines, but mainly for embedded system (automotive, industrial control, robotics, telecom, consumer electronics, etc.). Robert's expertise ranges from the smallest real-time systems (FreeRTOS) to set-ups with multiple processors/cores and embedded GNU/Linux (user-, kernel-space, device drivers, hardware interfacing, debugging, multi-core, Yocto Project) with a focus on free and open source software.

[Effective: February 2018]