

# The Yocto Project - A Summary



## Training Description

The Yocto-Project (YP) represents an Open-Source project which provides templates, tools and modes to generate Embedded Products independent of Linux based systems and their hardware architecture.

Even experienced GNU/Linux-users don't know how to accommodate their work flow to YP and whether they need it. This seminar will give answers and shows the most important elements which define the Yocto Project.

The seminar's aim is the imparting of the required essentials on base of existing knowledge in GNU and Linux.

As fundamental you should at least took the seminar Embedded Linux in theory and experienced. Also an advantage would be knowledge in GNU and/or Linux.

## Target Group

The workshop is aimed at software-, development-, system engineers, testers, administrators, engineers and other parties interested in the YP, with a solid knowledge of Embedded GNU/Linux.

## Prerequisites

- Basic familiarity with using a GNU/Linux system (e.g. Ubuntu) as an end user in user space
- Basic familiarity with a command line shell
- Basic knowledge of user/kernel space programming with GNU/Linux
- Intermediate C programming knowledge

## Training Content

### Day 1: Basics

- Introduction in the History of Unix/Linux, licenses and standards
- Embedded Specifics: Embedded Linux vs. Desktop Linux, Cross-/Native tool chains, Build-systems, C-libraries
- Eval Board
- Introduction in the Yocto Project
- The Yocto auto builder

## Day 2: The YP-Work flow and Bitbake

- Configurations and sources
- Building process: source fetching, patching, configure, compile, install, Pseudo, examples of recipes, Output Analysis/Packaging, image and SDK generation
- Customizing images: Intro, local.conf, IMAGEFEATURES, custom .bb files and package groups
- Bitbake's history and syntax
- Debugging:
  - Debug Level
  - Finding recipes/images/package groups
  - BitBake environment/tasks/logging
  - Build/force specific task cleansstate, stamp invalid explanation, Devshell
  - Dependence-Explorer
  - BitBake with graphic wrapper

## Day 3: Layers, Kernel and Application Development Toolkit (ADT)

- Layers Intro, BitBake-layers tool, yocto-layer tool
- Board Support Package (BSP) Intro, system development workflow, developer's handbook (bsp-tool)
- Kernel Intro, workflow system development, Kernel's developer handbook (defconfig, defconfig + configuration fragment, in tree kmod, out of tree kmod, fdt)
- Application Development Toolkit (ADT) Intro, Cross-Development tool chain, Sysroot, ADT-Eclipse Yocto plug-in, the QEMU emulator, user space tools
- Install ADT and tool chains (to use ADT-Installer Cross-tool chain tarball)

## Day 4: Debugging, Tracing and Profiling

- Debugging: gdb, gdb Remote-Debugging, (gdb Remote) Debugging with Eclipse, (remote) perform with Eclipse
- Tracing and Profiling: perf, gprof, gcov, strace, ftrace, systemtap, oprofile, LTTng + Eclipse (data visualization)
- Package management : working with packages, IPK, creating a package-feed, Installation of software suit with opkg on hardware
- Licensing, adding a customized license, Open-Source-License-Compliance
- Devtool, demonstrating the creation of meta-layers for a real project(meta-cfengine)

## Methods and Training Materials

Lecture and practical examples with host (laptops with Ubuntu 14.04.x LTS) and target system (z. B. Beagle Bone Black Rev.C - <http://beagleboard.org/BLACK>).

The devices will be provided during the training.

You receive a reference- and workbook, one Beagle Bone Black Rev. C plus one standard FTDI 3.3 V to USB cable. After the training you will get access to a download link with an already prepared Docker image and examples.

## Duration and Training Fee

4 days, 2.290 Euro plus VAT tax per person

This price includes a participants certification, the seminar documents, books and hardware, snacks as well as drinks and lunch.

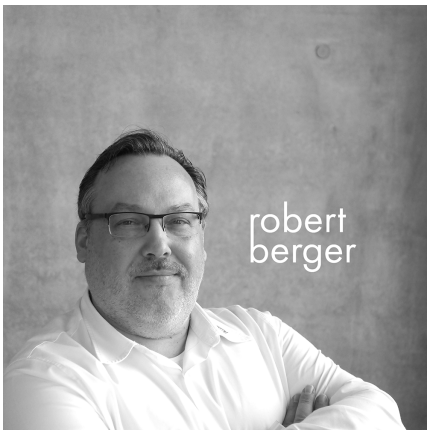
## Participants

6 to 12 participants

## Venue

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: May 2018]