Yocto Project and OpenEmbedded Training

A Yocto build your team owns: how BitBake reads the layers, and how to write the recipes, images, distro and machine layers of a product.

4 days	Intermediate	On site or remote	English or French
Duration	Level	Format	Language

Overview

This course covers the Yocto Project and OpenEmbedded from the first build to a custom distribution with its own BSP layer. It explains how BitBake reads its metadata, how recipes, classes and layers fit together, and how images, distributions and machines are described, so that a team can maintain its own product layers.

What You Will Learn

- Understand how BitBake parses metadata, resolves variables and runs tasks
- Write and debug recipes, classes and package variants
- Create custom images and a distribution layer with its own policies
- Write a BSP layer with its machine, kernel and bootloader configuration
- Provide an SDK to application developers
- Diagnose build failures with the development shell and devtool

Who should attend

Engineers who build or maintain an embedded Linux product with the Yocto Project, or who are about to move a product to it.

Prerequisites

Good knowledge of the Linux command line, and basic knowledge of embedded Linux: bootloader, kernel, root filesystem.

Program

The first two days cover BitBake and the metadata; the last two cover images, distributions, BSP layers and daily work.

Day 1

- Why build systems exist
- The Yocto Project, OpenEmbedded and Poky
- The build environment, configuration and running BitBake
- BitBake variables, assignment and override operators, inline Python

Day 2

- Advanced BitBake configuration: conditional syntax, providers and version preferences
- Layers and dynamic layers
- BitBake functions, logging and tasks
- Recipe fundamentals

Day 3

- Advanced recipes: package splitting, dependencies, variants, package groups, classes and features
- Creating and customizing images, wic images, custom image types
- The distribution layer: the policies a distro encodes, distro features, overrides

Day 4

- BSP layers and the machine concept: machine configuration, kernel, bootloader, multiconfig
- Application development: the SDK, sysroots and shared libraries
- Troubleshooting: reading build failures, the development shell, devtool

Labs and Hardware

Labs run on QEMU machines and on the boards of our lab: Arm 32-bit: STM32MP157C-DK2, BeagleBone Black; Arm 64-bit: Raspberry Pi 4 and 5, NXP i.MX8M Mini, Khadas VIM3; x86: QEMU x86, Generic x86_64 machines, Intel development kits. Other boards, or the hardware of the customer, can be used on request.

Sessions take place on site, in France, Belgium, Switzerland, Germany and beyond, with the hardware provided by LABCSMART, or remotely. Remote attendees work on our build server, each with their own SSH account and workspace, and run the labs on QEMU machines and on real boards. The exercises are drawn from our own client projects. Courses are given in English or French.

Optional Modules

These modules extend the course for specific products.

- x86 firmware and bootloaders in a Yocto build: UEFI, systemd-boot, GRUB, ACPI and microcode
- Container support in embedded systems and its Yocto integration
- Software updates with RAUC in a Yocto build

Who Teaches This Course

The course is designed and given by the founder of LABCSMART, an embedded Linux and Linux kernel engineer with more than eleven years of industrial experience. He contributes upstream to the Linux kernel for Renesas RZ SoCs, in the PCIe, audio, Ethernet, thermal, DMA and clock subsystems, and wrote two books published by Packt, *Linux Device Driver Development, Second Edition* (2022) and *Mastering Linux Device Driver Development* (2021), presented on our publications page.

Plan a session for your team. Every course is made of independent modules. A team can drop the modules it does not need and add others in their place, and we also build courses on demand around a given SoC, board or project. Contact: contact@labcsmart.com, https://labcsmart.com/company/contact/. Course page: https://labcsmart.com/training/yocto-project/