Zephyr RTOS Development Training

Real-time firmware written with Zephyr the way the project intends: devicetree, Kconfig, the driver model and portable application code.

5 days	Beginner to intermediate	On site or remote	English or French
Duration	Level	Format	Language

Overview

This course teaches how to build real-time applications and drivers with the Zephyr RTOS. It starts with the workspace and the build system, explains how the devicetree and the driver model describe and access hardware, goes through threads, interrupts and memory, and ends with the peripheral APIs and the sensor framework.

What You Will Learn

- Set up a Zephyr workspace and master the build system, Kconfig and overlays
- Describe hardware with the devicetree and write drivers for the Zephyr driver model
- Use threads, interrupts, synchronization and memory allocation correctly
- Structure an application with zbus, events and state machines
- Use I2C, SPI, UART, ADC, PWM, the sensor framework and persistent storage
- Debug with logging, the shell, core dumps and GDB

Who should attend

Firmware engineers starting with Zephyr, including those coming from bare-metal or another RTOS.

Prerequisites

Good knowledge of C and notions of microcontrollers.

Program

The program moves from the tools to the kernel, then to the hardware.

Day 1

- Zephyr, west and workspaces, toolchains
- The build system: configure and build phases, Kconfig, board targets and overlays, sysbuild, modules
- Utility macros, lists, ring buffers and atomic operations

Day 2

- Hardware description with the devicetree: bindings, generated macros, pin control
- The device driver model: initialization levels, device APIs, an out-of-tree driver

Day 3

- Threads, scheduling and interrupts
- Memory management: static allocation, slabs and heaps
- Structuring applications with zbus, events and state machines

Day 4

- Logging, shell and debugging, core dumps and stack analysis
- I2C, SPI and UART

Day 5

- ADC, PWM, watchdog and other peripherals
- The sensor framework
- Persistent storage: flash, NVS, ZMS, settings and file systems

Labs and Hardware

Labs run on STM32 Nucleo and Nordic nRF52840 DK boards. 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.

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/zephyr-rtos-development/