Fail-Safe OTA Updates Training for Embedded Linux: RAUC

Devices updated in the field without ever losing one, on eMMC as on raw NAND or NOR flash, and a whole fleet rolled out in phases with hawkBit.

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

Overview

This course shows how to update an embedded Linux device in the field without ever leaving it unbootable. It covers the failure cases an update must survive, A/B slots and boot counting, then the storage the update writes to: managed flash such as eMMC and SD cards does not behave like raw NAND or NOR flash, and the slot layout, the bootloader environment and the bootloader update all depend on it. The last day builds the complete update chain with RAUC in a Yocto image, and deploys it to a fleet of devices with Eclipse hawkBit.

What You Will Learn

- List the failure cases an update must survive
- Design an A/B or recovery update scheme with an atomic switch
- Implement boot counting and fallback in the bootloader
- Choose a slot layout for eMMC, NAND or NOR storage
- Update the bootloader itself without risking the device
- Sign update bundles, protect against downgrades and integrate RAUC into a Yocto image
- Roll updates out to a fleet with hawkBit, in phases and with monitoring

Who should attend

Engineers who design or integrate the update system of a connected embedded Linux product.

Prerequisites

Good knowledge of embedded Linux and basic knowledge of the Yocto Project.

Program

The first day covers the update mechanism, the second the storage it writes to, the third RAUC and the management of a fleet.

Day 1

- Updates as a security function
- What can go wrong during an update: forged images, power cuts, failed starts
- Update strategies: A/B slots, recovery systems, the atomic switch
- Boot counting and fallback in the bootloader
- Downgrades and anti-rollback

Day 2

- Managed flash (eMMC, SD) versus raw flash (NAND, NOR)
- MTD, UBI and UBIFS: wear leveling, bad blocks and power cuts
- Read-only root filesystems and the data partition
- Slot layouts per storage: GPT or MBR partitions, UBI volumes, MTD partitions
- A redundant bootloader environment
- Updating the bootloader: eMMC boot partitions, or redundant copies on NAND and NOR

Day 3

- RAUC: slots and slot types per storage, from raw and ext4 to UBI volumes, NAND, NOR and eMMC boot partitions
- Signed bundles, system configuration and Yocto integration
- Bundle hooks and handlers, boot confirmation
- Fleet management with Eclipse hawkBit: the server, rauc-hawkbit-updater on the device, device registration and attributes
- Target filters and groups, phased rollouts from a canary group to the whole fleet, monitoring of the update status
- Adaptive updates to cut download size

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.

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/embedded-linux-software-updates/