



STR23 - STM32MP2 Implementation

This course describe the STM32MP2 Implementation and practical examples

Objectives

- Understand the STM32MP2 architecture (dual Cortex-A35 + Cortex-M33) and memory/interconnect map.
- Bring up the full boot chain: TF-A → OP-TEE → U-Boot → Linux, and read meaningful boot logs.
- Configure and validate DDR and clock trees; confirm frequencies and governors on real hardware.
- Build and customize OpenSTLinux (Yocto) images; add a simple layer/recipe.
- Enable devices via Device Tree (pinmux, clocks, regulators) and debug with dmesg/sysfs.
- Set up storage and flashing flows (eMMC/SD/USB-DFU) with safe partitioning.
- Bring up key I/O: Ethernet/TSN, USB (host/gadget), and PCIe; run quick throughput checks.
- Configure graphics and media: DRM/KMS + GPU, VPU pipelines, CSI-2 camera and DSI/RGB display.
- Offload real-time work to the M33 core; exchange data with Linux via OpenAMP/RPMsg.
- Run a small Edge-AI sample (CPU/GPU/NPU) and compare performance/latency.
- Apply platform security: secure boot options, OP-TEE services, keys/OTP, tamper basics.
- Implement OTA/rollback strategy (SWUpdate/Mender concepts) and a practical recovery path.
- Tune power management (cpufreq/cpuidle) and record simple power profiles.
- Produce a production checklist: versioning, UID/serial, crash/reset logs, diagnostics, and update policy.

Day 3

Course Environment

- Theoretical course
 - PDF course material (in English) supplemented by a printed version for face-to-face courses.
 - Online courses are dispensed using the Teams video-conferencing system.
 - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
 - Practical activities represent from 40% to 50% of course duration.
 - Code examples, exercises and solutions
 - For remote trainings:
 - ▶ One Online Linux PC per trainee for the practical activities.
 - ▶ The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
 - ▶ QEMU Emulated board or physical board connected to the online PC (depending on the course).
 - ▶ Some Labs may be completed between sessions and are checked by the trainer on the next session.
 - For face-to-face trainings:
 - ▶ One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
 - ▶ One PC for two trainees when there are more than 6 trainees.
 - For onsite trainings:
 - ▶ An installation and test manual is provided to allow preinstallation of the needed software.
 - ▶ The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the trainee in his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
 - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
 - Quizzes are offered at the end of sections that do not include practical exercises to verify that the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
 - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites, in agreement with their company manager if applicable.

Plan**Day 1****MP2 SoC overview (A35/M33, interconnect)**

- Dual A35 cluster; M33 real-time core.
- AXI/AHB fabric; memory regions.
- GPU / VPU / NPU blocks (high-level).
- Display (RGB/LVDS/DSI), CSI-2 camera.
- High-speed I/O: PCIe, TSN GbE, USB

Exercise: SoC map quick tour

Boot chain & security foundation

- ROM & TF-A (FSBL) & OP-TEE (S-EL1) & U-Boot (SSBL) & Linux.
- Boot media: eMMC/SD/USB-DFU/NAND (board-dep).
- Device trees passed by U-Boot.
- Secure boot hooks & fuses (overview).
- Recovery & UART/USB consoles.

Exercise: Hello boot

DDR & clock/power bring-up

- DDR type (LPDDR4/DDR4/DDR3L) basics.
- TF-A DDR init vs board tuning.
- RCC & PLL trees; kernel/Peripheral clocks.
- DVFS/cpufreq overview.
- MCO/clock probes for validation

Exercise: Clock sanity

Pinmux & device-tree basics

- Linux pinctrl: groups/functions.
- DTS vs DT overlay structure.
- Regulators/GPIO/IRQ nodes.
- Clocks/resets in DT.
- Common mistakes & dmesg hints.

Exercise: DTS tweak

Storage & flashing

- Partitioning (boot/rootfs/data).
- eMMC vs SD vs NAND trade-offs.
- WIC/SD-card images vs dfu-util.
- U-Boot mmc/fatload/env tips.
- Filesystem choices (ext4/squashfs).

Exercise: Flash flow

Day 2

OpenSTLinux (Yocto) workflow

- Layers: meta-st-openstlinux, machine configs.
- repo init, bitbake targets.
- SDK/cross-toolchain export.
- Image types (core/weston/minimal).
- Incremental rebuilds & sstate.

Exercise: Build & boot

Kernel enablement

- Kconfig vs DTS responsibilities.
- Out-of-tree driver basics.
- Debug: dmesg, trace-cmd, devlink.
- UIO/GPIO-chardev access.
- Packaging modules in Yocto.

Exercise: LED/GPIO driver

Connectivity: Ethernet/TSN, USB, PCIe

- PHY setup; TSN capabilities (high-level).
- ip link, ethtool, iperf3 checks.
- USB gadget vs host roles.
- PCIe root complex basics.

Exercise: Net bring-up

Security & isolation (high-level)

- Resource isolation framework.
- TZC/ETZPC-like gates (periph/DDR).
- OP-TEE secure services.
- Keys/OTP and tamper (overview).
- Secure storage story

Exercise: OP-TEE ping

Day 3

Updates & recovery

- Dual-A/B rootfs strategy.
- SWUpdate/mender (concepts).
- U-Boot env for rollback.
- DFU recovery path.
- Factory vs field flows.

Exercise: Safe update sim

Graphics stack (DRM/KMS + GPU)

- DRM/KMS planes/connectors.
- 3D GPU overview (1080p UIs).
- Mesa/Wayland (Weston).
- Performance tips (buffers).
- Multi-display notes

*Exercise: Weston demo***Video pipeline (VPU)**

- Formats & pixel planes.
- H.264 decode/encode HW.
- GStreamer pipelines.
- Zero-copy paths idea.
- Storage & bandwidth tips

*Exercise: GStreamer test***Camera input (MIPI-CSI2)**

- CSI-2 lanes & DT nodes.
- Sensor drivers & controls.
- ISP blocks (conceptual).
- V4L2 capture basics.
- Sync/exposure notes.

*Exercise: Camera capture***Display outputs (RGB/LVDS/DSI)**

- Connector DT bindings.
- Mode setting (EDID/timings).
- Backlight/panel drivers.
- VSYNC/TE considerations.
- Tear & bandwidth checks

*Exercise: Panel bring-up***Edge AI accelerators (CPU/GPU/NPU)**

- NPU
- Toolchains (TFLite/ONNX).
- Delegate selection (CPU/GPU/NPU).
- Pre/post-processing tips.

*Exercise: Demo***Day 4****Cortex-M33 side (STM32CubeMP2)**

- CubeMP2 HAL/LL layout.
- Project templates in CubeIDE.
- Clocks/IRQs on M33.
- Bare-metal vs FreeRTOS.
- Debug attach options

*Exercise: M33 hello***A-core & harr; M-core IPC**

- VirtIO/RPMsg concepts.

- Shared memory carve-outs.
- Mailbox/interrupt routing.
- Message framing policy.

Exercise: RPMsg ping-pong

Field buses & timers (Linux + M33)

- UART/I²C/SPI from Linux.
- Offloading real-time to M33.
- PWM/capture (M33 side).
- DMA pipelines.
- Latency budgeting.

Storage, filesystems & performance

- eMMC/SD tuning (HS modes).
- ext4 vs squashfs trade-offs.
- I/O schedulers & fio.
- Wear leveling & logs.
- Mount options & journaling.

Exercise: I/O bench

Networking polish & security

- Firewalling basics (nftables).
- Secure shells/keys.
- Time sync (PTP/NTP).
- TSN concepts (sched/queues).
- Logging & metrics.

Exercise: Net hardening

Day 5

Power management

- cpufreq/governors.
- cpuidle/suspend-to-RAM.
- Peripheral low-power.
- Display blanking strategy.

Exercise: Power sweep

Robust boot & tamper

- OTP/fuses & boot keys.
- Active tamper inputs.
- Secure RTC/monitors.
- Rollback prevention.
- Audit logging

Exercise: Tamper demo

Production flashing & tests

- Golden image layout.
- Serial/UID/keys injection.
- Manufacturing scripts.
- Boundary/functional tests.
- Field diagnostics hooks.

Exercise: Factory script

Wrap-up & roadmap

- BSP update policy.
- Yocto release strategy.
- Security CVE handling.
- Backup & recovery docs.
- Next steps & references.

Exercise: Self-audit

Renseignements pratiques

Inquiry : 5 days