

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

Environnement du cours

- Cours théorique
 - Support de cours au format PDF (en anglais) et une version imprimée lors des sessions en présentiel
 - Cours dispensé via le système de visioconférence Teams (si à distance)
 - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
 - Les activités pratiques représentent de 40% à 50% de la durée du cours
 - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
 - Exemples de code, exercices et solutions
 - Pour les formations à distance:
 - ▶ Un PC Linux en ligne par stagiaire pour les activités pratiques, avec tous les logiciels nécessaires préinstallés.
 - ▶ Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique
 - ▶ Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante.
 - Pour les formations en présentiel:
 - ▶ Un PC (Linux ou Windows) pour les activités pratiques avec, si approprié, une carte cible embarquée.
 - ▶ Un PC par binôme de stagiaires s'il y a plus de 6 stagiaires.
 - Pour les formations sur site:
 - ▶ Un manuel d'installation est fourni pour permettre de préinstaller les logiciels nécessaires.
 - ▶ Le formateur vient avec les cartes cible nécessaires (et les remporte à la fin de la formation).
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours
- Au début de chaque session (demi-journée en présentiel) une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Modalités d'évaluation

- Les prérequis indiqués ci-dessus sont évalués avant la formation par l'encadrement technique du stagiaire dans son entreprise, ou par le stagiaire lui-même dans le cas exceptionnel d'un stagiaire individuel.
- Les progrès des stagiaires sont évalués de deux façons différentes, suivant le cours:
 - Pour les cours se prêtant à des exercices pratiques, les résultats des exercices sont vérifiés par le formateur, qui aide si nécessaire les stagiaires à les réaliser en apportant des précisions supplémentaires.
 - Des quizz sont proposés en fin des sections ne comportant pas d'exercices pratiques pour vérifier que les stagiaires ont assimilé les points présentés
- En fin de formation, chaque stagiaire reçoit une attestation et un certificat attestant qu'il a suivi le cours avec succès.
 - En cas de problème dû à un manque de prérequis de la part du stagiaire, constaté lors de la formation, une formation différente ou complémentaire lui est proposée, en général pour conforter ses prérequis, en accord avec son responsable en entreprise le cas échéant.

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 : RPSMsg 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

Renseignements : 5 jours