STR23 - Mise en œuvre du STM32MP2

Ce cours décrit la mise en œuvre du STM32MP2 et des exemples pratiques

Objectifs

- Comprendre l'architecture du STM32MP2 (double Cortex-A35 + Cortex-M33) et la cartographie mémoire et d'interconnexion.
- Mettre en route la chaîne de boot complète : TF-A vers OP-TEE vers U-Boot vers Linux, et savoir lire les traces de boot.
- Configurer et valider la DDR et les arbres d'horloges ; vérifier les fréquences et les governors sur la carte réelle.
- Construire et personnaliser des images OpenSTLinux (Yocto) ; ajouter une couche et une recette simples.
- Activer les périphériques via le Device Tree (pinmux, horloges, régulateurs) et déboguer avec dmesg et sysfs.
- Mettre en place le stockage et les chaînes de flashage (eMMC/SD/USB-DFU) avec un partitionnement sûr.
- Mettre en route les entrées-sorties clés : Ethernet/TSN, USB (host et gadget) et PCIe ; réaliser des contrôles rapides de débit.
- Configurer le graphisme et le multimédia : DRM/KMS et GPU, chaînes VPU, caméra CSI-2 et affichage DSI ou RGB.
- Déporter les traitements temps réel sur le cœur M33 ; échanger des données avec Linux via OpenAMP/RPMsg.
- Exécuter un petit exemple d'IA embarquée (CPU/GPU/NPU) et comparer performance et latence.
- Mettre en œuvre la sécurité de la plateforme : options de secure boot, services OP-TEE, clés et OTP, bases du tamper.
- Mettre en place une stratégie d'OTA et de rollback (concepts SWUpdate et Mender) et un chemin de reprise concret.
- Ajuster la gestion de l'énergie (cpufreq, cpuidle) et relever des profils de consommation simples.
- Établir une check-list de mise en production : versionnement, UID et numéro de série, journaux de crash et de reset, diagnostics et politique de mise à jour.

Troisième jour

Environnement du cours

- Cours théorique
  - Support PDF (en anglais), imprimé en présentiel ; à distance via Teams.
  - Assistance du formateur tout au long de la formation.
- Chaque session débute par un point avec les stagiaires.

Audience visée

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

Plan du cours

Premier jour

Présentation du SoC MP2 (A35/M33, interconnexion)

- Cluster double A35 ; cœur temps réel M33.
- Fabric AXI et AHB ; régions mémoire.
- Blocs GPU / VPU / NPU (vue d'ensemble).
- Affichage (RGB/LVDS/DSI), caméra CSI-2.
- Entrées-sorties haut débit : PCIe, TSN GbE, USB

Chaîne de boot et socle de sécurité

- ROM vers TF-A (FSBL) vers OP-TEE (S-EL1) vers U-Boot (SSBL) vers Linux.
- Supports de boot : eMMC/SD/USB-DFU/NAND (selon la carte).
- Device trees transmis par U-Boot.
- Points d'accroche du secure boot et fusibles (présentation).
- Reprise et consoles UART et USB.

Mise en route de la DDR, des horloges et de l'alimentation

- Bases du type de DDR (LPDDR4/DDR4/DDR3L).
- Initialisation DDR par TF-A et réglage propre à la carte.
- Arbres RCC et PLL ; horloges du noyau et des périphériques.
- Présentation du DVFS et de cpufreq.
- Sondes MCO et d'horloge pour la validation

Bases du pinmux et du device tree

- pinctrl sous Linux : groupes et fonctions.
- Structure des DTS et des DT overlays.
- Nœuds de régulateurs, de GPIO et d'interruptions.
- Horloges et resets dans le DT.
- Erreurs courantes et indices dans dmesg.

Stockage et flashage

- Partitionnement (boot, rootfs, données).
- Compromis entre eMMC, SD et NAND.
- Images WIC et cartes SD face à dfu-util.
- Conseils U-Boot : mmc, fatload, variables d'environnement.
- Choix de systèmes de fichiers (ext4, squashfs).

Deuxième jour

Chaîne de développement OpenSTLinux (Yocto)

- Couches : meta-st-openstlinux, configurations de machine.
- repo init, cibles bitbake.
- Export du SDK et de la chaîne de compilation croisée.
- Types d'images (core, weston, minimal).
- Reconstructions incrémentales et sstate.

Activation dans le noyau

- Responsabilités respectives du Kconfig et des DTS.
- Bases d'un driver hors arborescence.
- Débogage : dmesg, trace-cmd, devlink.
- Accès par UIO et par le chardev GPIO.
- Empaquetage des modules dans Yocto.

Connectivité : Ethernet/TSN, USB, PCIe

- Configuration du PHY ; capacités TSN (vue d'ensemble).
- Contrôles avec ip link, ethtool et iperf3.
- Rôles USB gadget et host.
- Bases du root complex PCIe.

Sécurité et isolation (vue d'ensemble)

- Framework d'isolation des ressources.
- Barrières de type TZC et ETZPC (périphériques et DDR).
- Services secure d'OP-TEE.
- Clés, OTP et tamper (présentation).
- Approche du stockage sécurisé

Troisième jour

Mises à jour et reprise

- Stratégie de rootfs double A/B.
- SWUpdate et Mender (concepts).
- Variables d'environnement U-Boot pour le rollback.
- Chemin de reprise par DFU.
- Flux en usine et sur le terrain.

Pile graphique (DRM/KMS et GPU)

- Plans et connecteurs DRM/KMS.
- Présentation du GPU 3D (interfaces 1080p).
- Mesa et Wayland (Weston).
- Conseils de performance (buffers).
- Notes sur le multi-écran

Chaîne vidéo (VPU)

- Formats et plans de pixels.
- Décodage et encodage H.264 matériels.
- Pipelines GStreamer.
- Principe des chemins zero-copy.
- Conseils sur le stockage et la bande passante

Entrée caméra (MIPI-CSI2)

- Lanes CSI-2 et nœuds de DT.
- Drivers de capteur et contrôles.
- Blocs ISP (conceptuel).
- Bases de la capture V4L2.
- Notes sur la synchronisation et l'exposition.

Sorties écran (RGB/LVDS/DSI)

- Bindings DT des connecteurs.
- Réglage des modes (EDID, chronogrammes).
- Drivers de rétroéclairage et de dalle.
- Prise en compte du VSYNC et du signal TE.
- Contrôles de tearing et de bande passante

Accélérateurs pour IA embarquée (CPU/GPU/NPU)

- NPU
- Chaînes d'outils (TFLite, ONNX).
- Choix du delegate (CPU/GPU/NPU).
- Conseils de pré-traitement et de post-traitement.

Quatrième jour

Partie Cortex-M33 (STM32CubeMP2)

- Organisation des couches HAL et LL de CubeMP2.
- Modèles de projet dans CubeIDE.
- Horloges et interruptions sur le M33.
- Bare-metal ou FreeRTOS.
- Options de rattachement du débogueur

IPC entre cœur A et cœur M

- Concepts VirtIO et RPMsg.
- Zones de mémoire partagée réservées.
- Routage des mailbox et des interruptions.
- Politique de découpage des messages.

Bus de terrain et timers (Linux et M33)

- UART, I²C et SPI depuis Linux.
- Déport du temps réel sur le M33.
- PWM et capture (côté M33).
- Chaînes DMA.
- Budget de latence.

Stockage, systèmes de fichiers et performance

- Réglage eMMC et SD (modes haute vitesse).
- Compromis entre ext4 et squashfs.
- Ordonnanceurs d'entrées-sorties et fio.
- Wear-leveling et journaux.
- Options de montage et journalisation.

Finalisation du réseau et sécurité

- Bases du pare-feu (nftables).
- Shells sécurisés et clés.
- Synchronisation du temps (PTP, NTP).
- Concepts TSN (ordonnancement, files).
- Journalisation et métriques.

Cinquième jour

Gestion de l'énergie

- cpufreq et governors.
- cpuidle et suspend-to-RAM.
- Basse consommation des périphériques.
- Stratégie d'extinction de l'écran.

Boot robuste et tamper

- OTP, fusibles et clés de boot.
- Entrées de tamper actif.
- RTC sécurisé et moniteurs.
- Prévention du rollback.

- Journalisation d'audit

Flashage de production et tests

- Organisation de l'image de référence.
- Injection du numéro de série, de l'UID et des clés.
- Scripts de fabrication.
- Tests aux limites et tests fonctionnels.
- Points d'accroche pour le diagnostic sur le terrain.

Synthèse et perspectives

- Politique de mise à jour du BSP.
- Stratégie de versions Yocto.
- Traitement des CVE de sécurité.
- Documentation de sauvegarde et de reprise.
- Prochaines étapes et références.