

## STR19 - STM32L5

*This course describes the STM32L5 architecture*

### Objectives

- Understand Cortex-M33 + TrustZone-M (secure vs non-secure).
- Partition memory/peripherals with SAU/IDAU and GTZC.
- Bring up Secure + Non-Secure projects and veneers (CMSE).
- Use RCC, GPIO/EXTI, timers/LPTIM, DMA/DMAMUX, ADC/COMP/OPAMP/DAC.
- Leverage crypto HW (AES/PKA/HASH/RNG) from Secure world.
- Apply low-power with TZ, RTC/tickless, and safe wake.
- Set up boot/Option Bytes (TZEN/RDP/PCROP) and basic TF-M / OEMiROT flow.
- Build a production checklist (watchdogs, reset logs, tamper).

### 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 ramène 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.

# Plan du cours

## Day 1

### Cortex-M33 + TrustZone-M

- Core Architecture (STM32 implementation options)
- Programmer's model
- Secure vs Non-Secure states.
- Exception targeting (S/NS).
- CMSE instructions (vener).
- MPU (S/NS regions).
- FPU (if present).

**Exercise :** Exception Management

**Exercise :** TrustZone Apps

**Exercise :** MPU v8

### STM32L5 SoC & memory map

- Flash/SRAM/PPB layout.
- IDAU regions (fixed).
- SAU adds S/NS windows.
- UID & Flash-size regs.
- TZEN Option Byte role.

**Exercise :** Map & IDs

### TrustZone-M partitioning (SAU/IDAU + veneers)

- SAU regions: S, NS, NSC.
- Veneer table (NSC funcs).
- Secure gateway calls.
- Faults on illegal access.
- Minimal service API design.

**Exercise :** NSC "get\_random()"

### GTZC security (mem & peripherals)

- MPCBB for SRAM/Flash.
- TZSC/TZIC periph gates.
- S/NS interrupt targets.
- DMA secure vs non-secure.
- Error flags & reports.

**Exercise :** Block NS access

### Secure/Non-Secure project bring-up

- CubeIDE dual-image setup.
- Separate vector tables.
- Startup order S & NS.
- Shared headers & ABI.
- Minimal NS app skeleton.

**Exercise :** Dual-image skeleton

## RCC - reset & clocks

- MSI/HSI16/HSE/PLL.
- SYSCLK/AHB/APB presc.
- CCIPR kernel clocks.
- MCO output, CSS.
- BOR/PVD quick notes.

**Exercise :** Clock profiles + MCO

## GPIO / EXTI with security

- S vs NS GPIO control.
- EXTI target (S/NS) lines.
- Debounce choices.
- SYSCFG & routing notes.

**Exercise :** Secure button, NS LED

## Day 2

## Timers & LPTIM (with TZ)

- PWM / capture / one-pulse.
- Encoder basics.
- Master/slave triggers.
- LPTIM for tickless wake.
- Secure timer services.

**Exercise :** Secure timebase

## DMA / DMAMUX (S vs NS)

- Channel/request map.
- Circular vs normal.
- HT/TC/TE handling.
- Coherency & barriers.
- Secure DMA policy.

**Exercise :** NS UART RX ring

## ADC / COMP / OPAMP / DAC

- Resolution & sampling.
- Oversampling option.
- Timer-triggered regular.
- DMA circular streaming.
- Watchdog & thresholds.

**Exercise :** ADC + DMA (NS)

## USART / LPUART / SPI / I<sup>2</sup>C (with TZ)

- UART 8/9-bit, parity.
- RX ring + idle detect.
- SPI CPOL/CPHA & DMA.
- I<sup>2</sup>C timeouts, bus-clear.
- S/NS peripheral policy.

**Exercise :** Comms (NS)

## Crypto accelerators (Secure)

- AES engine modes.
- HASH (SHA-1/256).
- PKA (ECC ops).
- RNG TRNG source.
- Key/nonce handling.

**Exercise :** Secure AES service

## Secure boot & TF-M (intro)

- ROM boot flow (L5).
- OEMiROT / SBSFU / TF-M.
- Image signing basics.
- NV counters / rollback.
- Logs & update hooks.

**Exercise :** Signed blink (demo)

## Third Day

## Low-power with TrustZone

- Sleep/Stop/Standby.
- Secure wake sources.
- SRAM/Reg retention.
- Re-init SAU on wake.
- Policy: who sleeps.

**Exercise :** Stop + secure wake

## RTC & tickless timing

- LSE vs LSI trade-offs.
- Calendar/alarm/wakeup.
- Backup registers.
- Tickless via LPTIM/RTC.

**Exercise :** Tickless blink (NS)

## USB FS device (variant)

- VBUS sense options.
- EP/FIFO sizing.
- CDC/DFU quick path.
- Clock constraints.
- Suspend/resume.

## Storage (QSPI/SDMMC) (variant)

- QSPI mapped reads.
- SDMMC + FatFS.
- Cache/ALIGN notes.
- Secure vs NS access.
- Basic file I/O test.

**Exercise :** SD + FatFS (NS)

## Option Bytes & production

- TZEN/RDP/PCROP/WRP.
- BOR/PVD levels.
- Tamper (TAMP) basics.
- Watchdogs IWDG/WWDG.
- Reset cause logging.

**Exercise :** OB snapshot & IWDG

### **Production checklist (wrap-up)**

- Partition doc (SAU/GTZC).
- Keys & secure services.
- Boot/update steps.
- Low-power numbers.
- UID/serial/CRC tags.

**Exercise :** Self-audit sheet