



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).

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.

Course Outline

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 / I²C (with TZ)

- UART 8/9-bit, parity.
- RX ring + idle detect.
- SPI CPOL/CPHA & DMA.
- I²C 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