



## STR18 - STM32 L4/L4+ implementation

*This course describe the STM32 L4/L4+ architecture and practical examples*

### Objectives

- Understand the Cortex-M4F core, DSP/FPU, and the L4/L4+ SoC.
- Configure RCC (MSI/HSI16/HSE/LSI/LSE) and PLL; validate prescalers.
- Use GPIO/EXTI, timers/LPTIM/RTC, DMA/DMAMUX, ADC/COMP/OPAMP/DAC.
- Apply ultra-low-power modes; measure current and wake latency.
- Bring up USART/SPI/I<sup>2</sup>C/SAI/DFSDM (where available).
- Use SDMMC + FatFS, QSPI/OSPI (variant), USB FS device.
- Manage Flash, Option Bytes, watchdogs, reset causes, and production checklists.

### 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.

# Course Outline

## Day 1

### Cortex-M4F overview (core)

- Programmer's Model
- Exceptions and NVIC priorities.
- DSP intrinsics (SIMD/MACC).
- FPU (single-precision).
- WFI/WFE sleep entry.
- ITM/SWO printf (if present).

**Exercise :** Exception Model

**Exercise :** Privilege Mode and Stack

### L4/L4+ SoC & memory map

- AHB/APB buses and bridges.
- Flash/SRAM/CCM (if any) layout.
- Peripheral address map.
- UID & Flash-size registers.
- Option Bytes snapshot.

**Exercise :** Map & IDs

### RCC - reset & clocks

- MSI ranges and auto-cal.
- HSI16 / HSE basics.
- PLLs.
- SYSCLK mux; AHB/APB divs.
- CCIPR kernel clocks.
- MCO output; CSS.

**Exercise :** Clock profiles

### Power & voltage scaling

- Voltage scaling VOS ranges.
- Low-power run/sleep.
- Stop / Standby modes.
- VBAT/backup domain.
- PVD/BOR levels.

**Exercise :** Mode sweep

### GPIO / EXTI / SYSCFG

- PP/OD, pulls, speed/drive.
- AF mapping principles.
- EXTI lines & priorities.
- Safe I/O at reset.

**Exercise :** EXTI + GPIO

### Timers & LPTIM

- PWM edge/center.

- Input capture, one-pulse.
- Encoder interface.
- Master/slave triggers.
- LPTIM for tickless.

**Exercise :** PWM + capture

## Day 2

### DMA / DMAMUX

- DMA1/DMA2 overview.
- Requests via DMAMUX.
- Normal vs circular.
- HT/TC/TE events.
- Restart & error paths.

**Exercise :** UART RX ring (DMA)

### ADC & triggers

- Resolution & sampling time.
- Oversampling & alignment.
- Timer-triggered regular.
- Injected group basics.
- DMA circular streaming.
- Analog watchdog.

**Exercise :** ADC + DMA stream

### OPAMP / COMP / DAC

- OPAMP PGA modes.
- COMP thresholds/hysteresis.
- Routing to timers/EXTI.
- DAC 12-bit with S&H.
- Calibrate & verify.

**Exercise :** Analog chain

### USART / LPUART

- USART/LPUART Overview.
- Oversampling 16/8.
- Blocking/IRQ/DMA.
- LPUART Stop-wake.
- Error handling (ORE/FE).

**Exercise :** Robust UART

**Exercise :** LPUART wake from stop

### SPI / I<sup>2</sup>S / SAI (variant)

- SPI CPOL/CPHA, NSS.
- Data sizes & duplex.
- DMA transfers.
- I<sup>2</sup>S basic audio path.
- SAI TX/RX blocks.

**Exercise :** SPI loopback DMA

### I<sup>2</sup>C

- Sm/Fm/Fm+ speeds.
- 7/10-bit addressing.
- Analog/digital filters.
- Timeouts; bus-clear.
- Clock stretching.

## Day 3

### RTC & tickless timing

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

**Exercise :** Tickless blink

### USB FS device (variant)

- VBUS sense options.
- EP/FIFO sizing basics.
- CDC/DFU quick paths.
- Clocking constraints.
- Suspend/resume flow.

**Exercise :** CDC echo or DFU

### Audio front-end (variant: DFSDM/SAI)

- DFSDM filter paths.
- Oversampling & decimation.
- SPI/I<sup>2</sup>S microphones.
- DMA to circular buffer.
- Basic RMS meter.

**Exercise :** DFSDM stream

### Storage & external memory

- SDMMC + FatFS basics.
- QSPI mapped mode.
- OSPI (L4+ variants).
- Buffer align/invalidate.
- File-I/O sanity tests.

**Exercise :** SD + FatFS

## Day 4

### Low-power measurement

- GPIO leakage strategy.
- Kernel clocks for LP.
- Stop entry/exit timing.
- DMA + LP coexistence.
- Practical power tips.

**Exercise :** Measure Sleep/Stop

### Security & integrity (variant)

- RNG TRNG.
- AES/HASH engines.
- PKA (L4R/L4S).
- CRC for images/packets.
- Key storage basics.

**Exercise :** CRC/AES

## **Flash, OB & robustness**

- Flash erase/program.
- EEPROM emulation.
- OB: RDP/PCROP/BOR.
- Watchdogs IWDG/WWDG.
- Reset cause logging.

## **Boot, ROM & production checklist**

- ROM bootloader ports.
- DFU/UART/I<sup>2</sup>C options.
- Boot pins & OB ties.
- Version/UID/CRC tags.
- Final audit list.

**Exercise :** Self-audit