

## STR13 - STM32U5

*This course describe the STM32U5 architecture*

### Objectives

- Understand STM32U5 architecture (Cortex-M33, clocks, resets, memory, peripherals).
- Configure low-power features and measure real consumption.
- Implement drivers (GPIO, timers, UART/I<sup>2</sup>C/SPI, ADC, DMA) with interrupts.
- Set up secure boot basics and TrustZone-M partitioning (overview + hands-on).
- Build and debug a small, robust application integrating peripherals, low power, and (optionally) RTOS.

### Prerequisites

- Familiarity with C concepts and programming targeting the embedded world
- Prior MCU experience (any STM32 helpful).
- Related:
  - cours [RT3 - Programmation temps réel avec FreeRTOS](#)
  - cours [RT5 - Programmation avec Zephyr RTOS](#)
  - cours [L2 - C language for Embedded MCUs](#)
  - cours [STR9 - STM32 Peripherals](#)

### 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

### ARM Cortex-M33 overview

- Programmer's model, stacks, privilege levels.
- Exceptions & NVIC, SysTick, fault handling (practical tips).
- Memory protection overview (MPU), TrustZone-M concept (intro).

### STM32U5 architecture overview

- Block diagram, buses, resets.
- Memory map (Flash, SRAM), caches/buffers (device-specific), option bytes.
- RCC: clock sources, PLL, prescalers; safe re-clocking patterns.

### GPIO & EXTI

- Power pins
- Pinout
  - Pin Muxing, alternate functions
- GPIO Module
  - Configuring a GPIO
  - Analog function
  - Integrated pull-up / pull-down
  - I/O pin multiplexer and mapping
  - TrustZone security

**Exercise :** Configure EXTI for a button (rising/falling), ISR and debouncing.

## Day 2

### DMA / LPDMA & DMAMUX

- Roles of DMA vs CPU, request lines, bursts, circular/normal modes
- LPDMA specifics for ultra-low-power transfers.
- DMAMUX request mapping and overrun handling.

**Exercise :** DMA-GPIO heartbeat

**Exercise :** LPDMA ADC stream

**Exercise :** DMAMUX remap test

### General-purpose Timers (PWM, Input Capture)

- Prescalers, ARR, CCR; PWM modes and dead-time basics.
- Input capture for frequency/period; filtering/glitch removal.
- One-pulse mode and master/slave timer chaining.

**Exercise :** Generate PWM on a timer channel; sweep duty cycle with a button.

**Exercise :** Frequency meter using input capture; print measured Hz.

### Low-power Timers (LPTIM) & RTC

- LPTIM vs GPTIM, LSE/LSI sources and accuracy.
- RTC calendar/alarms, sub-second, backup domain care.
- Wakeup sources and Stop/Standby interplay.
- Timestamping and drift considerations.

**Exercise :** LPTIM periodic wake from Stop

**Exercise :** RTC alarm wake + backup register persistence check

## Communications (UART, I<sup>2</sup>C, SPI)

- USART
  - Modes & framing
  - DMA & flow control
  - Errors & diagnostics
- I<sup>2</sup>C
  - Master transfers
  - Bus management & recovery
  - Robustness
- SPI
  - Modes & timing
  - DMA & chip-select
  - Integrity & performance

**Exercise :** UART DMA

**Exercise :** Periodic sensor

**Exercise :** SPI demo

## ADC

- Triggers & sampling times; oversampling
- DMA to ring buffer; window statistics
- Internal channels (V<sub>ref</sub>, temperature)
- Noise sources & layout tips

**Exercise :** Timer-triggered ADC + DMA

## Storage (optional) - SDMMC + FatFS

- Card detect & init; mount/format
- File append patterns; buffering
- Latency & wear considerations
- Safe close on power loss

**Exercise :** Log “timestamp, ADC” to CSV

## PWR & Low-power modes

- Low-power modes overview
- Wake sources; retention/autonomous peripherals
- VOS scaling; SMPS/LDO notes
- BOR/PVD/PVM supervision

**Exercise :** Sleep vs Stop current table

**Exercise :** Practical low-power measurements

## Day 3

### Boot modes & FLASH Option Bytes

- Boot sources & vector relocation
- Key OBs incl. NS/S boot address
- RDP overview & implications
- Read/verify OBs safely

### TrustZone & GTZC

- SAU/IDAU concepts; NS/S partitioning
- GTZC: TZSC/TZIC/MPCBB roles
- Peripheral/memory isolation basics
- Enabling/disabling TZ

**Exercise :** S + NS projects

**Exercise :** secure veneer call

**Exercise :** demo NS access fault &rarr; wrapper

## MPU & Privilege

- Region types & no-exec guards
- Privileged vs unprivileged access
- Fault status registers & context capture
- Fail-safe patterns

**Exercise :** MPU fault and logs

## Crypto & Secure storage

- SAES engine (features overview)
- OTFDEC for external/XIP content
- Unique ID & key management basics
- Watermarks/HDP (high-level)

**Exercise :** Secure RNG service from S to NS

## Access-controlled debug & Life-cycle

- Debug vs RDP levels
- Product life-cycle states (brief)
- Provisioning principles (keys/placeholders)
- Rollback/unlock procedures

**Exercise :** Provisioning dry-run and restore board to training state

## Updates (IAP / dual-image)

- Bootloader/app split; CRC/hash check
- “Update pending” flags & rollback
- Trigger paths (command/flag/comms)
- Jump sequence & vector table remap

## Robustness & resets

- IWDG vs WWDG; service windows
- BOR levels; startup implications
- Reset flags: POR/WDG/SW/Standby
- Minimal reset log at boot

**Exercise :** Inject hang &rarr; IWDG reset; print last reset cause on boot

## Tracing & logging

- ITM/SWO quick setup
- Event markers around ISRs/DMA
- Timestamped printf (lightweight)
- Buffering vs blocking

**Exercise :** ITM printf: visualize DMA callback markers