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/I²C/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:
  - RT3 - FreeRTOS Real Time Programmingcourse
  - RT5 - Zephyr RTOS Programmingcourse
  - L2 - C language for Embedded MCUscourse
  - STR9 - STM32 Peripheralscourse

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.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

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, I²C, SPI)

- USART
  - Modes & framing
  - DMA & flow control
  - Errors & diagnostics
- I²C
  - 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 (Vref, 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*

Renseignements pratiques

Inquiry : 3 days