



## STR9 - STM32 Peripherals

*This course describe the STM32 family peripherals (STM32Fx, STM32Lx and STM32MPx)*

### Objectifs

- Describing the different peripherals of the STM32 family of 32-bit Flash microcontrollers based on the ARM Cortex-M processor
  - The Ultra-Low-Power (STM32 L0, STM32 L1, STM32 L4, STM32 L4+ and STM32 L5)
  - The Main stream (STM32 F0, STM32 F1, STM32 F3 and STM32 G0)
  - The High Performance (STM32 F2 STM32 F4, STM32 F7 and STM32 H7)
- This course also cover the STM32MP series peripherals
- Getting started with the ST Drivers to program STM32 peripherals (The STM32Cube Library):
  - Configuring the peripheral using CubeMX
  - Accessing the peripheral through the HAL and LL libraries
- Note: some complex peripherals, which are only accessed through the ST-provided drivers and some middleware stack, are not described in this course but in more specific courses:
  - The Ethernet (ETH) media access control MAC with DMA controller, covered in the [STS1 - LwIP Implementation](#) course or [STG - STM32 + FreeRTOS + LwIP](#) course
  - The USB controllers, covered in the [IP2 - USB 2.0](#) course
  - The CAN controller, covered in the [IA1 - CAN bus](#) course
  - The LCD-TFT controller, handled through emWin, covered in the [STG - STM32 + FreeRTOS + LwIP](#) course

### Course environment

- Example code, labs and solutions are provided to the attendees.
- Labs are done on STM32 boards using System Workbench for STM32 and CubeMx

### Prerequisites and related courses

- Familiarity with C concepts and programming targeting the embedded world
- Basic knowledge of embedded processors specially the ARM Cortex-M
- The following courses could be of interest:
  - [STS1 - LwIP Implementation](#) course
  - [IA1 - CAN bus](#) course
  - [IP2 - USB 2.0](#) course and related specifications: OTG 3.0, xHCI, UAS and AV classes
  - [STR4 - STM32 F0-Series implementation](#) course
  - [STR5 - STM32 F1-Series implementation](#) course
  - [STR6 - STM32 F2-Series implementation](#) course
  - [STR7 - STM32 F4-Series implementation](#) course
  - [STR8 - STM32MP15 Implementation](#) course

### Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

# Course Outline

## First day

### STM32 series Overview

- STM32 Ultra-low-power series architecture overview
  - STM32L0, STM32L1, STM32L4, STM32L4+, STM32L5
- STM32 main stream architecture overview
  - STM32F0, STM32G0, STM32F1, STM32F3
- STM32 high performance architecture overview
  - STM32F2, STM32G0, STM32F1, STM32F3
- STM32 MPU microprocessor
  - STM32MP153, STM32MP157

### STM32 Core architecture overview

- ARM v7-M core family
- Core Architecture
- Programming
- Exception behavior
- Basic interrupt operation, micro-coded interrupt mechanism
- Floating point unit, DSP instructions and MPU
- ARM Cortex-M low power modes

### Power and Clock Management

- Reset and Clock Control (RCC)
  - Overview
  - RCC block diagram
  - RCC Reset
  - RCC Clock
  - RCC interrupts
  - RCC Applications

**Exercise:** RCC Clock Configuration

### Real-time clock (RTC)

- Introduction and main features
- Functional description
  - Clock and prescalers
  - Real time clock and calendar
  - Programmable alarms
  - Periodic auto-wakeup
  - Initialization and configuration
  - Reading the calendar
  - Resetting the RTC
  - Synchronization
  - RTC reference clock detection
  - RTC coarse digital calibration
  - RTC smooth digital calibration
  - TimeStamp function
  - Tamper detection
  - Calibration clock output

- Alarm Output
- RTC and low-power modes
- RTC interrupts

**Exercise:** RTC Alarm

**Exercise:** RTC Calendar

**Exercise:** RTC Time Stamp

## Embedded Flash memory interface

- Introduction
- Main features
- Embedded Flash memory in STM32 series
- Read Interface
  - Relation between CPU clock frequency and Flash memory read time
  - Adaptive real-time memory accelerator (ART Accelerator)
- Erase and program operation
- Option bytes
- One time programmable bytes
- Flash interface registers

## Flexible Memory Controller (FMC)

- FMC main features
- Block diagram
- AHB interface
- External device address mapping
- NOR Flash / PSRAM controller
- NAND Flash / PC Card Controller
- SDRAM Controller
  - SDRAM Controller main features
  - SDRAM External memory interface signals
  - SDRAM controller functional description
  - Low Power modes
  - SDRAM controller registers

## Second day

## Flexible Static Memory Controller (FSMC)

- FSMC main features
- Block diagram
- AHB interface
- External device address mapping
  - NOR / PSRAM address mapping
  - NAND / PC Card address mapping
- NOR Flash / PSRAM Controller
  - External memory interface signals
  - Supported memories and transactions
  - General timing rules
  - NOR flash / PSRAM controller asynchronous transactions
  - Synchronous transactions
- NAND Flash / PC card Controller
  - External memory interface signals
  - NAND Flash / PC Card Supported memories and transactions
  - Timing diagrams for NAND and PC Card
  - NAND Flash operations
  - NAND Flash prewait functionality

- Computation of the error correction code (ECC)
- PC Card / CompactFlash operations

**Exercise:** FSMC SRAM basic functionalities

**Exercise:** FSMC SRAM data memory

## Direct Memory Access (DMA) Controller

- DMA STM32 Series
- DMA introduction
- DMA main features
- DMA Functional Description
  - DMA Transactions
  - Channel selection
  - Arbiter
  - DMA streams
  - Source destination and transfer modes
  - Pointer incrementation
  - Circular mode
  - Double buffer mode
  - Programmable data width, packing/unpacking, endianness
  - Single and burst transfers
  - FIFO
  - DMA transfer completion
  - DMA transfer suspension
  - Flow Controller
  - Stream configuration procedure
  - Error Management
- DMA Interrupts
- Using the STM32F2, STM32F4 and STM32F7 DMA controller
- Using the STM32F0/F1/Lx DMA Controller

**Exercise:** DMA FIFO mode

**Exercise:** DMA FLASH to RAM

## Chrom-Art Accelerator Controller (DMA2D)

- DMA2D overview
- DMA2D functional description
  - DMA2D control
  - DMA2D foreground and background FIFOs, pixel format converter (PFC) and CLUT interface
  - DMA2D blender
  - DMA2D output PFC and FIFO
  - DMA2D AHB master port timer
  - DMA2D transactions and configuration
  - DMA2D transfer control
- DMA2D interrupts
- Using the DMA2D to refresh an LCD-TFT Display on the STM32L4
- Embedded graphics on STM32F4

## Analog-to-Digital Converter (ADC)

- STM32 ADC capabilities
- ADC introduction and main features
- ADC functional description
  - ADC on-off control
  - ADC clock
  - Channel selection
  - Single conversion mode
  - Continuous conversion mode

- Timing diagram
- Analog watchdog
- Scan mode
- Injected channel management
- Discontinuous mode
- Data alignment
- Channel wise programmable sampling time
- Conversion on external trigger and trigger polarity
- Fast conversion mode
- Data management using DMA
- Multi ADC modes
  - Injected simultaneous mode
  - Regular simultaneous mode
  - Interleaved mode
  - Alternate trigger mode
  - Combined regular/injected simultaneous mode
  - Combined regular simultaneous +alternate trigger mode
- Temperature sensor
- Battery charge monitoring
- ADC interrupts
- ADC differences between the STM32 series

**Exercise:** ADC dual mode Interleaved

**Exercise:** ADC Injected Conversion interrupt

**Exercise:** ADC Regular Conversion DMA

**Exercise:** ADC Regular Conversion interrupt

**Exercise:** ADC Regular Conversion Polling

**Exercise:** ADC Trigger Mode

**Exercise:** ADC Triple Mode Interleaved

## Third day

### Digital-to-analog Converter (DAC)

- STM32 DAC capabilities
- DAC introduction and main features
- DAC functional description
  - Channel and output buffer enable
  - DAC data format
  - DAC conversion
  - DAC output voltage
  - DAC trigger selection
  - DMA request
  - Noise generation
  - Triangle wave generation
- Dual DAC channel conversion
- DAC capabilities in The STM32 families

**Exercise:** DAC Signals Generation

**Exercise:** DAC Simple Conversion

### Advanced-Control Timers

- Timers capabilities in The STM32 families
- Introduction and main features
- Functional description
  - Time-base unit
  - Counter modes
  - Repetition counter

- Clock selection
- Capture and compare channels
- Input capture mode
- PWM input mode
- Forced output mode
- Output compare mode
- PWM mode
- Complementary outputs and dead time insertion
- Using break function
- 6-step PWM generation
- One pulse mode
- Encoder interface mode
- Timer input XOR function
- Interface with Hall sensors
- TIMx and external trigger synchronization
- Timer synchronization
- Debug Mode
- Timers capabilities in the STM32 series

**Exercise:** TIM Complementary Signals

**Exercise:** TIM DMA example

**Exercise:** TIM DMA burst

**Exercise:** How to configure TIM1 TIM1 peripheral in encoder mode to determinate the rotation direction

**Exercise:** TIM External Trigger Synchronization

**Exercise:** TIM Input Capture

## General-Purpose timers

- Introduction and main features
- Functional description
  - Time-base unit
  - Counter modes
  - Clock selection
  - Input capture mode
  - PWM input mode
  - Forced output mode
  - Output compare mode
  - PWM mode
  - One pulse mode
  - Encoder interface mode
  - Timer input XOR function
  - Timers and external trigger synchronization
  - Timer synchronization
- STM32 General purpose Timers

**Exercise:** TIM Cascade Synchronization

**Exercise:** Configuring the TIM peripheral to generate four different signals with four different delays

## Basic Timers

- Introduction and main features
- Functional description
  - Time-base unit
  - Counting mode
  - Clock source
  - Debug mode
- STM32 Basic Timers

**Exercise:** TIM 6 Steps

**Exercise:** TIM 7 PWM Output

## Fourth day

### Independent and Window Watchdog (IWDG / WWDG)

- IDWG
  - Introduction and main features
  - Hardware watchdog
  - Register access protection
  - Debug mode
  - Registers
- WWDG
  - Introduction and main features
  - Functional description
  - How to program the watchdog timeout
  - Debug mode
- IDWG/WWDG capabilities in the STM32 series

**Exercise:** Independent Watchdog

### Cryptographic processor (CRYP)

- Cryptographic processor in the STM32 series
- Introduction and main features
- CRYP functional description
  - DES/TDES cryptographic core
  - AES cryptographic core
  - Initialization vectors
  - CRYP busy state
  - Procedure to perform an encryption or a decryption
  - Context swapping
- Interrupts
- DMA Interface
- CRYP capabilities in the STM32 series

**Exercise:** CRYP AES Mode

**Exercise:** CRYP AES DMA

**Exercise:** Encrypt and Decrypt data using DES and TDES Algorithms

**Exercise:** Encrypt data using TDES Algorithm in ECB mode with DMA

### Random number generator (RNG) and Hash processor (HASH)

- RNG and Hash Processor in the STM32 series
- Random number generator
  - Introduction and main features
  - Functional description
- Hash processor
  - Introduction and main features
  - Functional description
- RNG and Hash Processor capabilities in the STM32 series

**Exercise:** Multiple Random Number Generator

**Exercise:** HMAC digest calculation using HMAC SHA1 and HMAC MD5

**Exercise:** HASH digest calculation using SHA1 and MD5 (with DMA)

### Universal Synchronous Asynchronous Receiver Transmitter (USART)

- USART introduction
- USART in the STM32 series
- USART functional description

- USART character description
- Transmitter
- Receiver
- Fractional baud rate generation
- USART receiver tolerance to clock deviation
- Multiprocessor communication
- Parity Control
- Local interconnection Network (LIN) Mode
- USART synchronous mode
- Single-wire half duplex communication
- SmartCard
- IrDA SIR ENDEC block
- Continuous Communication using DMA
- Hardware flow control
- USART interrupts
- USART mode configuration
- USART capabilities in the STM32 series

**Exercise:** UART printf

**Exercise:** UART Hyperterminal IT

**Exercise:** UART Hyperterminal DMA

## Secure digital input/output interface (SDIO)

- SDIO main features
- SDIO bus topology
- SDIO in the STM32 series
- DIO functional description
- Card Functional description
- Response formats
- SDIO I/O card-specific operations
- CE-ATA specific operation
- HW flow control
- SDIO capabilities in the STM32 series

## Fifth day

### Inter-integrated Circuit I2C features

- I2C introduction
- I2C in the STM32 series
- Functional description
  - Mode selection
  - I2C slave and master mode
  - Error conditions
  - Programmable noise filter
  - SDA/SCL line control
  - SMBus
  - DMA request
  - Packet error checking
- I2C interrupts
- I2C Debug mode
- I2C capabilities in the STM32 series

**Exercise:** I2C two boards Advanced Communication IT/DMA

**Exercise:** I2C Two Boards Communication Polling

**Exercise:** Multiple I2C data buffer transmission/reception between two boards in interrupt mode with restart condition

## Serial Peripheral Interface (SPI)

- SPI overview
- SPI and I2S in the STM32 series
- SPI functional description
  - General description
  - Configuring the SPI in slave or master mode
  - Configuring the SPI for half-duplex communication
  - Data transmission and reception procedures
  - CRC calculation
  - Status flags
  - Disabling the SPI
  - SPI communication using DMA
  - Error flags
  - SPI interrupts
- I2S functional description
  - General description
  - I2S full duplex
  - Supported audio protocols
  - Clock generator
  - I2S master and slave mode
  - Status flags
  - Error flags
  - I2S interrupts
  - DMA capability
- I2S/I2C capabilities in the STM32 series

**Exercise:** Transmit / Receive SPI data buffer using Interrupt, in an advanced communication mode

**Exercise:** Perform SPI data buffer transmission/reception between two boards via DMA

## Serial audio interface (SAI)

- Introduction and main features
- Block diagram
- Main SAI modes
- SAI synchronization mode
- Audio data size
- Frame synchronization
- Slot configuration
- SAI clock generator
- Internal FIFOs
- AC'97 link controller
- Specific features
  - Mute mode
  - MONO/STEREO function
  - Comanding mode
  - Output data line management on an inactive slot
- Error flags
- Interrupt sources
- Disable the SAI
- SAI DMA interface
- SAI capabilities in the STM32 series

## Digital Camera interface (DCMI)

- DCMI in the STM32 series
- DCMI introduction and main features
- DCMI pins

- DCMI clocks
- DCMI functional overview
  - DMA interface
  - DCMI Physical interface
  - Synchronization
  - Capture modes
  - Crop features
  - JPEG format
  - FIFO
- Data format description
  - Monochrome format
  - RGB format
  - YCbCr format
- DCMI interrupts
- DCMI capabilities in the STM32 series

**Exercise:** DCMI Capture Mode

**Exercise:** DCMI Snapshot Mode