



AAM - ARM Cortex-M Architecture (v7/v8)

This course explains the ARM Cortex-M global architecture.

Objectives

- Describing the ARM Cortex-M processors architecture (ARMv6, ARMv7 and ARMv8)
- Presenting the Hardware and Software implementation possibilities to learn how to create Cortex-M based applications
- Review the differences between the different Cortex-M cores
- Get an overview of the new features included in ARMv8 architecture (TrustZone, MPU, New Memory Types, ...)
- Learn how to:
 - Develop and debug an ARM Cortex-M application
 - Configure and Manage Exceptions/Interrupts
 - How to configure a privileged and unprivileged access with the MPU
- This course provides all the prerequisites for the courses describing in details the various Cortex-M cores and CPUs.

Course environment

- Theoretical course
 - PDF course material (in English)
- For online courses :
 - Course dispensed using the Teams video-conferencing system
 - The trainer to answer trainees' questions during the training and provide technical and pedagogical assistance through the Teams video-conferencing system
 - One Online Linux PC per trainee for the practical activities
 - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.

Prerequisites

- Familiarity with embedded C concepts and programming
- Basic knowledge of embedded processors

Target Audience

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

Course Outline

First Day

ARM v6M/v7M/v8M Architecture Overview

- Introduction to the ARM Architecture
- Cortex-M Processors
- Programmers' Model
 - Core Registers
 - Privileges, Modes and Stacks
 - Instruction Set
 - Datapath and pipeline, speculative branch target prefetch
- Exception Model
- Memory Model

- Address Map
- Memory Types
- Instruction and data alignment
- System Control Space
- Power Management

Exercise: Core register review and changing Mode and Stack

Exercise: Using low power mode

Cortex-M Implementation Diversity

- ARM Cortex-M0 processor
- ARM Cortex-M0+ processor
- ARM Cortex-M3 processor
- ARM Cortex-M4 processor
- ARM Cortex-M7 processor
- ARM Cortex-M33 processor
- ARM Cortex-M23 processor
- ARM Cortex-M processor family comparison

Second Day

Cortex-M Software Development

- Tools
 - Keil IDE and Ulink2 probe presentation
 - System Workbench for MCU
- Standards
 - AAPCS
 - UAL
 - CMSIS
- Code Generation
 - Variable types supported
 - Register Usage, Parameter passing
 - Aligned and Unaligned accesses
 - Endianness
- Image Generation
 - Dealing with branches
- Fault Tolerance
 - Stack issues
- Determinism
- RTOS Support
 - MPU Overview
 - SysTick Timer Overview

Exercise: AAPCS review, CMSIS utilization and Assembly language inlining

Cortex-M Optimization

- Compiler optimizations
 - Optimization levels
 - Tail-call
 - Inlining of function
 - Loop transformation
 - Multifile compilation
 - Floating point
- Bit Banding
- Memory copy optimizations
- Base pointer optimization

Exercise: Bit banding implementation

Cortex-M Debug

- ARMv6-M and ARMv7-M Debug Overview
 - Coresight presentation
- Invasive Debug
 - Breakpoints and Watchpoints
 - Vector Catch
 - Semi-hosting
- Non-invasive Debug
 - Data Watchpoint and trace unit
 - Instrumentation Trace Macrocell (ITM)
 - Embedded Trace Macrocell (ETM)
 - Micro Trace Buffer (MTB)

Exercise: Debug features review through the Keil IDE

Cortex-M Startup and Linker

- Reset Behavior
 - Vector Table
- CMSIS-CORE Startup and System Initialization
 - Startup File
 - Exception Handlers
 - Stack and heap setup
- Post Startup Initialization
- Working with the linker
 - Creating code and data sections
 - Placing code and data in memory

Exercise: Startup sequence to the main() review

Exercise: Executing the code from a SRAM

Third Day

Cortex-M Exception Model

- Exception Handling
- Exception entry and exit
- Exception stacking
- Nesting
- Tail-chaining
- Late-arriving

Exercise: Exception entry review

- Prioritization and Control
- NVIC registers
- Priority boosting
- Priority grouping
- Masking exceptions
- Writing Interrupt Handlers
- Interrupt Sensitivity
- Internal Exceptions and Faults
- Fault escalation

Exercise: Managing interrupts and priorities

Memory Protection Unit (MPU) ARMv7 and ARMv8

- MPU regions
- Privileged vs Unprivileged

- Memory Types
- Access permissions
- Region overlapping
- Access Fault

Exercise: MPU Utilization

Fourth Day

Cortex-M TrustZone

- Security States
- Register Banking
- Secure State Address Protection
- Secure and Non-Secure Interactions

Cortex-M Advanced Features

- SysTick Configuration and Calibration

Exercise: Working with the SysTick Timer

- Synchronization
 - Critical section, atomicity
 - LDREX/STREX instructions
 - Lock and unlock examples
- Memory Barriers
 - Data memory, Data Synchronization and Instruction Synchronization Barriers
 - Utilization examples
- Further Instruction Set Information
 - Return on the Instruction Set
 - If-Then Block
 - DSP extension Overview

Exercise: Using DSP instructions