



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

Ce cours explique l'architecture globale ARM Cortex-M

Objectifs

- Description de l'architecture des processeurs ARM Cortex-M (ARMv6, ARMv7 et ARMv8).
- Passer en revue les différences entre les différents noyaux Cortex-M.
- Obtenez un aperçu des nouvelles fonctionnalités incluses dans l'architecture ARMv8 (TrustZone, MPU, nouveaux types de mémoire, ...).
- Apprendre à :
 - Développer et déboguer une application ARM Cortex-M
 - Configurer et gérer les exceptions/interruptions
 - Comment configurer un accès privilégié et non privilégié avec le MPU.
- Ce cours fournit tous les prérequis pour les cours décrivant en détail les différents cœurs et CPU Cortex-M.

Environnement du cours

- Cours théorique
 - Matériel de cours en PDF (en anglais)
- Pour les cours en ligne :
 - Cours dispensé via le système de visioconférence Teams
 - Le formateur répond aux questions des stagiaires pendant la formation et fournit une assistance technique et pédagogique par le biais du système de visioconférence Teams.
 - Un PC Linux en ligne par stagiaire pour les activités pratiques.
 - Le formateur a accès aux PCs en ligne des stagiaires pour l'assistance technique et pédagogique.

Pré-requis

- Familiarité avec les concepts et la programmation du C embarqué
- Connaissance de base des processeurs embarqués

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Plan du cours

Premier jour

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

Deuxième jour

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

Troisième jour

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

Quatrième Jour

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