Training ARM Cortex-M Architecture: This course explains the ARM Cortex-M global architecture. - Processors: ARM Cores

AAM - ARM Cortex-M Architecture

This course explains the ARM Cortex-M global architecture.

Objectives

- Describing the ARM Cortex-M processors architecture
- 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
- This course provides all the prerequisites for the courses describing in details the various Cortex-M cores and CPUs.

Course environment

- Convenient course material with space for taking notes
- Demos targeting Cortex-M4/M33 based MCU using the GCC6based AC6 System Workbench IDE

Prerequisites

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

Plan

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

Exercice: *Core register review and changing Mode and Stack*
Exercice: *Using low power mode*

Cortex-M Implementation Diversity

- ARM Cortex-M0 processor
- ARM Cortex-M0+ processor
- ARM Cortex-M3 processor
- ARM Cortex-M4 processor
- ARMv6-M versus ARMv7-M

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

Exercice: *AAPCS review, CMSIS utilization and Assembly language inlining*

Second Day

Cortex-M Optimization

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

Exercice: *Bit banding implementation*

Cortex-M Debug

- ARMv6-M and ARMv7-M Debug Overview
  - Coresight presentation
- Invasive Debug
  - Beakpoints 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)

Exercice: *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

Exercice: *Startup sequence to the main() review*

Exercice: *Executing the code from a SRAM*

Third Day

Cortex-M Advanced Features

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

Exercice: *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

Exercice: *Managing interrupts and priorities*

- Memory Protection Unit (MPU)
  - Initialization
  - Memory types and access permissions
  - Region overlapping

Exercice: *MPU Utilization*

- SysTick Configuration and Calibration

Exercice: *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

Exercice: *Using DSP instructions*

Using a GCC Toolchain

Exercice: *Creating a Cortex-M based application using a GCC toolchain (with AC6 System Workbench for MCU)*

Renseignements pratiques

Duration : 3 days
Cost : 2000 € HT