Cortex-M0 Software Development

Course Description

Cortex-M0 software development is a 2 days ARM official course.
The course goes into great depth and provides all necessary know-how to develop software for systems based on Cortex-M0 processor.

The course covers the Cortex-M0 architecture, development tools, instruction set, interrupt handling, memory management, C programming and debug.

At the end of the course the participant will receive a certificate from ARM.

Course Duration

2 days

Goals

1. Become familiar with ARMv6-M architecture
2. Become familiar with Cortex-M0 architecture
3. Become familiar with ARMv6-M instruction set
4. Become familiar with the development tools for Cortex-M
5. Be able to handle interrupts
6. Understand the memory structure in v6-M architecture
7. Write an efficient C code for Cortex-M processor
8. Be able to debug your design
9. Be able to write software for Cortex-M microcontrollers

Target Audience

Software engineers that would like developing software for platforms based on Cortex-M0 microcontroller.

Prerequisites

- Computer architecture background
- C and Assembler
- Experience in developing embedded systems

Course Material

- ARM official course book
- Labs handbook
- Keil MDK-ARM

Agenda

Main Topics:

- Introduction to the ARM Architecture
- Cortex M0 Architecture Overview
- Tools Overview for ARM Microcontrollers
- V6-M Programmer's Model
- V6-M Compiler Hints & Tips
- V6-M Memory Model
- V6-M Exception Handling
- Embedded Software Development for Cortex-M Processors
- V6-M Linker & Libraries Hints & Tips
- Cortex-M0 Debug

Day #1

- **The ARM Architecture**
  - Embedded & applications processor roadmap
  - Introduction to the ARM architecture

- **Cortex-M0 Core**
  - Cortex-M0/M3/M4(F) architecture overview
  - Programmer's model
  - Pipeline
  - Memory map
  - Bit-banding
  - System timer (SysTick)
  - Alignment and Endianness
  - System control block

- **Tools Overview for ARM Microcontrollers**
  - Introduction to Keil MDK
  - ULINK debug adapters
  - Development boards
  - DS5 and DSTREAM
  - Fast Models from ARM

- **v6-M Programmer's Model**
  - ARMv6-M profile overview
  - Data types

- Core registers
- Modes, privilege and stacks
- Exceptions
- Instruction set overview

- **v6-M C/C++ Compiler Hints & Tips**
  - Basic compilation
  - Compiler optimizations
  - Coding considerations
  - Mixing C/C++ and assembler
  - Local and global data issues

Day #2

- **v6-M Memory Model**
  - Introduction to Cortex-M memory model
  - Memory address space
  - Memory types and attributes
  - Endianness
  - Barriers
  - System caches and TCMs

- **v6-M Exception Handling**
  - Exception architecture overview
  - Exception model
  - Interrupts handling
  - Interrupts prioritization and control
  - Writing the vector table and interrupts handlers in C/C++ and assembly
  - Internal exceptions and RTOS support
  - Fault exceptions
  - Interrupt sensitivity

- **Embedded Software Development for Cortex-M Processors**
  - Embedded development process
  - An “out-of-the-box” build
  - Tailoring the C library to your target
  - Tailoring image memory map to your target
  - Reset and initialization
  - Further memory map considerations
  - Building and debugging your image
  - Placing stack and heap in scatter file

- **v6-M Linker & Libraries Hints & Tips**
  - Linking basics
  - System and user libraries
  - Veneers
  - Stack issues
  - Linker optimizations and diagnostics
  - ARM supplied libraries
  - Scatter-loading

- **Cortex-M0 Debug**
  - Debug overview
  - What is CoreSight?
  - Invasive and non-invasive debug
  - Debug Access Port (DAP)
  - Breakpoints/watchpoints & vector catch
  - Cortex-M0 debug
  - System control