Cortex-M3/M4 Software Development

Course Description

Cortex-M3/M4 software development is a 3 days ARM official course.
The course goes into great depth and provides all necessary know-how to develop software for systems based on Cortex-M3/M4 processor.

The course covers the Cortex-M3/M4 architecture, development tools, instruction set, interrupt handling, memory management, migration process from older ARM microcontrollers, C programming, using MPU, debug, floating point and DSP instructions.

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

Course Duration

3 days (4 days with hands-on labs)

P.O.B. 803 Kfar Saba 441080 ISRAEL | P +972-52-5816791 | F +972-77-4702742 | ContactUs@HandsOnTraining.co.il | HandsOnTraining.co.il

Goals

1. Become familiar with ARMv7-M architecture
2. Become familiar with Cortex-M3/M4 architecture
3. Become familiar with ARMv7-M instruction set
4. Become familiar with the development tools for Cortex-M
5. Be able to handle interrupts
6. Be able to configure and use the MPU
7. Understand the memory structure in v7-M architecture
8. Write an efficient C code for Cortex-M processor
9. Be able to debug your design
10. Become familiar with DSP and FP instructions
11. Be able to write software for Cortex-M microcontrollers

Target Audience

Software engineers that would like developing software for platforms based on Cortex-M3/M4 microcontroller.

Prerequisites

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

Course Material

ARM official course book
Labs handbook
Keil MDK-ARM

P.O.B. 803 Kfar Saba 441080 ISRAEL | P +972-52-5816791 | F +972-77-4702742 | ContactUs@HandsOnTraining.co.il | HandsOnTraining.co.il

Agenda

Main Topics:

- Introduction to the ARM Architecture
- Cortex M3/M4 Architecture Overview
- Tools Overview for ARM Microcontrollers
- v7-M Programmer's Model
- v7-M Assembly Programming
- v7-M Memory Model
- v7-M Exception Handling
- Software Engineer's Guide to Cortex-M3/M4
- v7-M Compiler Hints & Tips
- v7-M Linker & Libraries Hints & Tips
- Migrating Legacy ARM/Thumb Code to Cortex-M3/M4
- Embedded Software Development for Cortex-M Processors
- Cortex-M3/M4 Debug
- Cortex-M3/M4 MPU
- Cortex-M3/M4 DSP and SIMD Instructions
- Cortex-M4 Floating Point Instruction Set

Day #1

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

- **Cortex-M3/M4 Core**
  - Cortex-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

- **v7-M Programmer's Model**
  - ARMv7-M profile overview
  - Data types
  - Core registers
  - Modes, privilege and stacks
  - Exceptions
  - Instruction set overview

- **v7-M Assembly Programming**
  - Why do you need to know assembler?
  - Data processing instructions
  - Load/Store instructions
  - Flow control
  - Miscellaneous instructions

Day #2

- **v7-M Memory Model**
  - Introduction to Cortex-M memory model
  - Memory address space
  - Memory types and attributes
  - Alignment and endianness
  - Barriers
  - System caches and TCMs

- **v7-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

- **Software Engineer's Guide to Cortex-M3/M4**
  - Cortex-M3/M4 instruction set

- Modes and stacks
- Special purpose registers
- System memory interface
- SysTick timer
- Cortex-M4F FPU
- Power management
- CMSIS

- **v7-M C/C++ Compiler Hints & Tips**
  - Basic compilation
  - Compiler optimizations
  - Coding considerations
  - Mixing C/C++ and assembler
  - Local and global data issues

- **v7-M Linker & Libraries Hints & Tips**
  - Linking basics
  - System and user libraries
  - Veneers
  - Stack issues
  - Linker optimizations and diagnostics
  - ARM supplied libraries
  - Scatter-loading

Day #3

- **Migrating Legacy ARM/Thumb Code to Cortex-M3**
  - ARM7TDMI vs. Cortex-M3
  - Converting assembler
  - Compiling C/C++ code
  - Thumb2 vs. Thumb

- **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

- **Cortex-M3 Debug**
  - Basic debug requirements

P.O.B. 803 Kfar Saba 441080 ISRAEL | P +972-52-5816791 | F +972-77-4702742 | ContactUs@HandsOnTraining.co.il | HandsOnTraining.co.il

- CoreSight architecture overview
- Debug events and reset
- Flash patch and breakpoint unit (FPB)
- Data watchpoint and trace unit (DWT)
- Instrumentation trace and ETM
- Physical interfaces

- **Cortex-M3/M4 Memory Protection**
  - Why do we need memory management?
  - Access permissions
  - Types & attributes
  - Memory Protection Unit (MPU)

- **Cortex-M3/M4 DSP, SIMD, and FP Instructions**
  - Differences between Cortex-M3 and Cortex-M4
  - Cortex-M4 DSP and SIMD instructions
  - Cortex-M4 optional floating point overview
  - FPU instruction set
  - FPU exception stack frame