1-800-633-1440
www.mindshare.com
training@mindshare.com

ARM MCU Architecture

Training

Let MindShare Bring “ARM MCU Architecture” To Life For You

The ARM MCU Architecture course focuses on software aspects of the ARMv6-M and ARMv7-M Architecture profiles (Cortex-M). This course is aimed at embedded software and systems developers who wish to acquire a broad knowledge of ARM technology with a bias toward the microcontroller market.

This course was endorsed by ARM for anyone wanting to take the ARM MCU Certification exam when ARM was still offering their AAME certification program (AAME - ARM Approved MCU Engineer). That program has been discontinued by ARM, but the information in this course is still very relevant and extremely valuable for anyone working with ARM microcontrollers.

ARM Endorsement

“On the strength of MindShare’s position as an industry leader in technology training, particularly in the area of microprocessor architecture, we were happy to welcome them to the AAE program as an ARM Accreditation Training Partner (AATP). Their expertise in delivering engaging self-paced eLearning courses should make MindShare's ARM Accredited MCU Engineer Certification eLearning course a very good way to efficiently and cost-effectively prepare for the ARM Accredited MCU Engineer (AAME) exam”.
- Daniel Dearing, AAE Program Manager

You Will Learn:

- ARM processor architecture (M0/M0+/M3/M4)
- ARM architecture v6-M/v7-M registers
- ARM instructions, syntax and assembly language programming
- Interrupt handling environment
- v6-M/v7-M memory model and protection
- Embedded software development and C/C++ compiler hints
- Cortex-M test and debug

Course Length: 3 Days

Course Outline:

- ARM Introduction
  - Discusses the contents of the course and provides an intro to ARM the company
- Cortex-M3 / M4 Overview (v7-M)
  - Cortex-M3 / M4 block diagram, feature set, M-Profile instructions and cycle counting
  - Overview of register set, processor pipeline, memory map, bit-banding, modes, privileges, stacks, interrupts and exceptions, power management, implementation options
- Cortex-M0 / M0+ Overview (v6-M)
  - Overview of register set, processor pipeline, memory map, interrupts and exceptions, power management, RTL configuration options, debug options, system interfaces
- Tools Overview for ARM Microcontrollers
  - Keil MDK components, ULINK debug adapters, development boards, DSTREAM debug and trace unit, fast models, ARM tools licensing, legacy tools: RVDS, RVI, RVT(2), MPS
- ARMv6-M and ARMv7-M Programmers' Model
  - Integer register set, Cortex-M4 FP extensions, Program Counter (PC), Link Register (LR), Stack Pointer (SP), v7M and v6M Program Status Register (PSR), Privileged vs unprivileged execution, exception handling, v7M vs v6M instruction set, instruction classes: branch instructions, data processing instructions, load and store instructions,

1-800-633-1440
www.mindshare.com
training@mindshare.com

status register access instructions, and more, conditional execution

- CMSIS (Cortex Microcontroller Software Interface Standard)
  - CMSIS structure, CMSIS-CORE, CMSIS-DSP, CMSIS-RTOS, CMSIS-SVD, CMSIS-Pack, CMSIS-Driver, CMSIS-DAP
- Cortex-M3 / M4 Microarchitecture
  - Processor block diagram, pipeline details (fetch, decode, execute), several pipeline examples with instructions, instruction folding, LDR and STR examples, write buffer
- Cortex-M0 Microarchitecture
  - Processor block diagram, prefetch buffer, hardware multiplier, pipeline details, several pipeline examples with instructions, execution determinism, instruction cycle timing
- Cortex-M0+ Microarchitecture
  - Processor block diagram, prefetch buffer, hardware multiplier, pipeline details, several pipeline examples with instructions, IO Port, execution determinism, instruction cycle timing
- Assembler Programming for ARMv7-M Processors
  - Instruction set basics, Unified Assembler Language (UAL), assembly file explanation, numerous C to assembly examples, operand options
  - Multiply, divide, bit manipulations, memory accesses, pre- vs. post-indexing, using the stack, If-Then blocks, condition codes and flags, exclusive access instructions, multi-thread mutex, power management instructions, Thumb-2 improvements, converting legacy assembler
- ARMv7-M Memory Model
  - System address map, memory segments, Private Peripheral Bus (PPB), System Control Space (SCS), System Control Block (SCB), memory types and properties, instruction and data alignment, endianness, memory barrier instructions (DMB, DSB, ISB), system caches and caching behavior, normal memory vs device memory, memory access ordering
- ARMv6-M Memory Model
  - System address map, memory segments, Private Peripheral Bus (PPB), System Control Space (SCS), System Control Block (SCB)
- ARMv7-M Exception Handling
  - Exception overhead, exception types, external interrupts, exception model, vector table, exception priorities, interrupt entry and exit timing, NMI, nesting, tail chaining
  - Priority boosting, disabling interrupts, priority groups, interrupt control and status fields, pulse vs level-sensitive interrupts, writing interrupt handlers, System Service Call (SVC), fault exceptions, lockup state, precise vs imprecise exceptions
- ARMv6-M Exception Handling
  - Vector table, exception priorities, returning from interrupt, NMI, supported priority groups, disabling interrupts, interrupt control and status bits, fault exceptions, lockup state
- C / C++ Compiler Hints and Tips
  - Variable types supported, optimization levels, automatic optimizations, use of "volatile", instruction scheduling, inlining of functions, loop unrolling, branch target optimizations, multi-file compilation
  - Register usage, parameter passing, loop termination, division operations, floating point, Cortex-M3 / M4 bit-banding, C++ support, mixing C and Assembly, CMSIS, intrinsics, named register variables, embedded assembler, inline assembler, variable types, global data, packing of structures, alignment of pointers, optimization of memcpy(), base pointer optimization
- Linker and Library Hints and Tips
  - Purpose of linker, object files, libraries, creating and maintaining libraries, scatter-loading, veneers, stack issues, unused section elimination, raw data compression, small function inlining, linking for specific target, debug issues, useful linker diagnostics
- ARMv7-M Synchronization
  - Need for atomicity, critical sections, LDREX and STREX instructions, lock() and unlock() examples, context switching, memory attributes, multiprocessor mutex, synchronization primitives, legacy SWP instruction

- Embedded Software Development for Cortex-M Processors
  - Embedded development process, default memory map, default C library, reset and initialization, CMSIS startup and initialization, scatter-loading, linker placement rules, root regions
  - Run-time memory management, stack and heap, MPU initialization, extending functions, Thumb C libraries, retargeting the C library, debugging ROM images
- Cortex-M3 / M4 Debug and Trace
  - Debug features, CoreSight overview, debug access paths, halted debug mode, vector catch, semihosting, Flash Patch and Breakpoint unit (FPB), Data Watchpoint and Trace (DWT), Instrumentation Trace Macrocell (ITM), Embedded Trace Macrocell (ETM), TPIU interface, physical interfaces
- Cortex-M0 Debug
  - CoreSight, non-invasive debug, debug state, debug events, Debug Access Port (DAP), ROM Tables, breakpoints and watchpoints, Breakpoint Unit (BPU), Data Watchpoint and Trace (DWT), vector catch, debug vs non-debug version, registers and fields related to debug
- Cortex-M0+ CoreSight Micro Trace Buffer
  - CoreSight MTB-M0+, MTB interfaces, MTB operation, MTB packet format, memory model, POSITION register, MASTER register, FLOW register, BASE register, using TSTART and TSTOP signals, AHB-Lite interface, SRAM memory interface, trace enable and disable sequence
- ARMv7-M Memory Protection
  - Memory map and protection overview, MPU and bus faults, regions, sub-regions, memory types, access permissions, region overlapping (overlaying), MPU-related registers
- Cortex-M4 Additions
  - Cortex-M4 features, PPA, gate count summary, DSP capability and instruction set, SIMD instructions, single precision floating point

Recommended Prerequisites:

Knowledge of basic computer architecture is recommended.

Course Material:

Students will be provided with an electronic version of the slides used in class.