

## **Cortex®-M7, M4, M3, M0+, M0: Arm® Cortex-M Architektur Training - Präsenz-Training**

### **Ziele - Ihr Nutzen**

Sie kennen die Cortex-M7, M4, M3, M1, M0 Architektur und können Programme in Assembler und C erstellen. Sie können die Programme im Speicher platzieren und testen. Sie haben den perfekten Einstieg in die Entwicklung von Cortex-M-basierenden Systemen.

### **Teilnehmer**

Hardware- und Software-Entwickler

### **Voraussetzungen**

ANSI-C und Mikrocontroller-Grundkenntnisse.

## **Cortex®-M7, M4, M3, M0+, M0: Arm® Cortex-M Architektur Training - Präsenz-Training**

### **Inhalt**

#### **Cortex®-M (Armv7-M, Armv6-M) Prozessor-Architektur**

- Register-Organisation, Special Purpose Register
- Operation Modes (Handler/Thread, privileged/unprivileged)
- Main Stack, Process Stack
- Cortex™-M Pipelinekonzept
- Cortex™-M Memory Map, System Control Block, Bit Banding

#### **Überblick über die Arm Prozessor Cores**

- Cortex®-M, Cortex®-R, Cortex®-A
- Arm7/9/10/11

#### **Cortex®-M7, M4, M3, M0+, M0 Instruction Set**

- Thumb-2 Instruction Set
- Data Processing Instructions
- Branch and Control Flow Instructions, Subroutines
- Branch Table, If ... then Conditional Blocks
- Data Access Instructions
- Memory Barriers and Synchronization
- Exclusive Access Primitives
- Assembler-Direktiven
- Praktische Übungen zur Erstellung kleiner Assembler-Routinen und zum Debuggen

#### **Exception und Interrupt Handling**

- Exception Model
- Reset, NMI, Faults, SysTick, Debug, Supervisor Calls, External Interrupts
- Tail Chaining, Late Arriving
- Nested Vector Interrupt Controller (NVIC)
- Interrupt Configuration and Status
- Interrupt Prioritization, Priority Grouping
- Fault Handler
- Praktische Übungen zum SystemTick, Supervisor Call und PendSV im Kontext von RTOS-Anwendungen
- Praktische Übungen zu den Fault Handlern und Ausgabe von Status-Informationen

#### **Reset Modes, Clock Generation, Power Management**

- Clock Generation

- Resets und Cortex®-M Reset Modes
- Sleep Modes und Power Management
- System Timer

**Memory Protection Unit MPU für Embedded Systeme**

- Armv6-M und Armv7-M MPU
- Statische Konfiguration der MPU
- Dynamische Umprogrammierung der MPU im RTOS-Kontext
- Praktische Übungen zur Anwendung der MPU

**Cache, Tightly Coupled Memory (TCM)**

- Cache-Grundlagen
- Caches und TCM des Cortex®-M7
- Cache-Konfiguration über die MPU

**Embedded Core Debugging**

- Core und System Debugging
- JTAG Debug Port
- 2-Pin Single Wire Debug Port
- Trace Port Interface Unit
- Embedded Trace Macro Cell
- Praktische Übungen zum Debuggen von C-Code mit dem µVision-Debugger und Printausgaben auf die Debug-Konsole

**Embedded Software Development**

- Bibliotheksroutinen an die Hardware anpassen (Retargeting)
- Code und Daten im Speicher platzieren (Scatter Loading)
- Linker Description File
- Processor Startup, Startup File
- Praktische Übung zur Platzierung von Code und Daten an vordefinierten Adressen

**Effiziente C-Programmierung für die Cortex-Architektur**

- Compiler-Optimierung, Compiler-Optionen
- Schnittstelle C - Assembler
- Programmierrichtlinien für Cortex-Compiler
- Lokale und globale Daten optimal verwenden

**Hardwarenahe C-Programmierung nach CMSIS**

- Cortex Mikrocontroller Software Interface Standard (CMSIS)
- Softwarearchitektur für Embedded-Systeme
- Strukturierte Beschreibung von Peripherie
- Zugriff auf Peripherie in C
- C-Statements und deren Ausführung in Assembler
- Praktische Übung zur Nutzung der CMSIS-Funktionen, z.B. zur Programmierung des NVIC Interrupt Controllers

**Floating Point Unit, Digital Signal Processing**

- Architekturüberblick zur FPU
- Exception Handling mit FPU
- Single-Instruction Multiple Data (SIMD) und Saturation Befehle

**Überblick Cortex®-M (Armv8-M und Armv8.1-M) Prozessor-Architektur**

- Einführung in die Armv8-M Prozessorarchitektur
- Erweiterungen der Armv8.1-M Prozessorarchitektur (HELIUM)
- Unterschied zur Armv6-M und Armv7-M Prozessorarchitektur
- Änderungen der neuen Armv8-M MPU
- Überblick zu Cortex®-M23, M33 und Arm TrustZone

**Übungen mit Keil µVision in Assembler und C**

- Praktische Übungen zu Armv7-M Cortex-M7 werden auf dem STM32F746 Nucleo-Board entwickelt und getestet.
- Die Übungen werden mit Keil Studio (Visual Studio Code) durchgeführt. Keil uVision wird manchmal als Debugger verwendet.

**MicroConsult Plus**

- Sie erhalten von uns Ihre Übungsverzeichnisse und Lösungsbeispiele für alle Übungsaufgaben.
- Zusätzlich erhalten Sie eine Installationsanleitung mit Download-Links der Toolumgebung, um die Übungen

auch nach dem Training nachvollziehen zu können.

-----

### **Präsenz-Training**

<b>Termin</b>	<b>Preis *</b>	<b>Dauer</b>
17.11.2026 – 20.11.2026	2.800,00 €	4 Tage

\* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: CORMX

### **Live-Online - Deutsch**

<b>Termin</b>	<b>Dauer</b>
20.07. – 23.07.2026	4 Tage

### **Präsenz-Training - Englisch**

<b>Termin</b>	<b>Dauer</b>
17.11. – 20.11.2026	4 Tage

### **Live-Online - Englisch**

<b>Termin</b>	<b>Dauer</b>
20.07. – 23.07.2026	4 Tage

### **Coaching**

Unsere Coaching-Angebote bieten den großen Vorteil, dass unsere Experten ihr Wissen und ihre Erfahrungen direkt in Ihren Lösungsprozess einbringen und damit unmittelbar zu Ihrem Projekterfolg beitragen.

Für Ihre Anfrage oder weiterführende Informationen stehen wir Ihnen gern zur Verfügung.