MICRO CONSULT

Stand 20.05.2026

Cortex®-R4, R5, R7, R8: Arm® Cortex-R Architektur Training - Präsenz-Training

**Ziele -
Ihr Nutzen**

Sie kennen die Cortex® R4, R5, R7 und R8 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-basierenden Systemen.

Teilnehmer

Software- und Hardware-Entwickler. (Sollten Sie bereits das Training "Arm7/9/10/11: Architektur und Embedded-Programmierung" besucht haben, setzen Sie sich bitte vorab mit uns in Verbindung).

Voraussetzungen

ANSI-C und Mikrocontroller-Grundkenntnisse.

Cortex®-R4, R5, R7, R8: Arm® Cortex-R Architektur Training - Präsenz-Training

Inhalt

Arm Cortex Prozessor-Architektur
- Register-Organisation, Operation Modes, States, Pipeline

Arm Prozessor Cores Übersicht
- Cortex®-R4, -R5, -R7, -R8 Prozessor-Core
- Cortex®-Av7, Cortex™-Av8 Prozessor-Cores
- Cortex®-Mv7, Cortex™-Mv8 Prozessor-Cores
- Arm7/9/11 Prozessor-Core

Arm, Thumb, Thumb-2 Instruction Sets
- Arm v4, v4T, v5, v6 Instruction Set
- Thumb Instruction Set
- v7 Thumb-2 Instruction Set
- Data Barriers, Instruction Barriers
- Synchronization, Load/Store Exclusive Instructions
- Arm/Thumb Interworking
- Assembler-Direktiven

Exception Handling
- FIQ, IRQ, Abort, Supervisor Call, Undefined
- Exception Handler Examples
- Vectored Interrupt Controller, VIC
- Generic Interrupt Controller, GIC

Coprocessors, Floating Point Unit
- Arm Coprozessor-Konzept
- System Controller CP15
- Floating Point Unit, FPU
- Systemkonfiguration

L1 Memory Interface
- Tightly Coupled Memory, Cache-Architektur
- Memory Protection Unit, MPU

L2 Memory Interface
- AXI, Advanced Microprocessor Bus Architecture

- Master Interface
- Slave Interface

Debug, Trace, Performance Monitoring

- Watchpoint Units, Embedded Trace Macrocell ETM
- Performance Monitor Unit, PMU

Multi-Processing Features

- Private Memory Region
- Snoop Control Unit, SCU
- Hardware Coherency Management
- Split Mode (Performance Mode)
- Locked Mode (Safety Mode)

Power Modes

- Run, Standby, Dormant, Shutdown

Embedded Software Development

- Bibliotheksroutinen an die Hardware anpassen (Retargeting)
- Code und Daten im Speicher platzieren (Scatter Loading)
- Linker Description File
- Reset, Startup, Startup File
- Von Reset bis Main

Effiziente C-Programmierung für die Cortex-Architektur

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

Hardwarenahes C

- C-Statements und deren Ausführung in Assembler
- Zugriff auf Peripherie in C
- Schichtenmodell für Embedded-Systeme
- Strukturierte Beschreibung von Peripherie

Übungen mit der Keil µVision und den Arm RealView Tools

- Auf Anfrage können auch weitere Tools eingesetzt werden
- Alle Programme werden auf einem Evaluierungsboard getestet

Präsenz-Training

Preis *	Dauer
-	4 Tage

Anmeldecode: CORRX
* Preis je Teilnehmer, in Euro zzgl. USt.

Live-Online - Deutsch

Dauer

4 Tage

Präsenz-Training - Englisch

Dauer

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.