

Cortex®-M23, M33: Armv8-M Architektur Training für Teilnehmer mit Kenntnissen der Vorgänger-Version - Live-Online-Training

Sie lernen die neue ARMv8-M Architektur (Cortex®-M23 und -M33) kennen und können Programme in Assembler und C erstellen. Der Schwerpunkt dieses Workshops liegt auf Software und deckt zahlreiche Themen ab, wie z.B. TrustZone, Prozessorarchitektur, erweiterter Befehlssatz, Interruptverhalten uvm. Nach dem Training können Sie die Programme in gemischter Secure- und Non-Secure-Konfiguration im Speicher platzieren und testen. Sie sind perfekt vorbereitet für die Entwicklung von Cortex®-M23/M33-basierenden Systemen.

Ziele - Ihr Nutzen

Sie lernen die neue ARMv8-M Architektur (Cortex®-M23 und -M33) kennen und können Programme in Assembler und C erstellen.

Der Schwerpunkt dieses Workshops liegt auf Software und deckt zahlreiche Themen ab, wie z.B. TrustZone, Prozessorarchitektur, erweiterter Befehlssatz, Interruptverhalten uvm.

Nach dem Training können Sie die Programme in gemischter Secure- und Non-Secure-Konfiguration im Speicher platzieren und testen.

Sie sind perfekt vorbereitet für die Entwicklung von Cortex®-M23/M33-basierenden Systemen.

Teilnehmer

Das Armv8-M Architektur-Training richtet sich an Hardware- und Software-Entwickler, die Grundkenntnisse der Vorgänger-Architektur haben (Armv6-M / Armv7-M Architektur des Cortex®-M0/M0+/M3/M4 oder -M7).

Voraussetzungen

Grundkenntnisse der Armv6-M / Armv7-M Architektur des Cortex®-M0/M0+/M3/M4 oder -M7 sind unbedingt erforderlich. Ebenso werden ANSI-C und Mikrocontroller-Grundkenntnisse vorausgesetzt. Das Training konzentriert sich auf die neuen Möglichkeiten des Cortex®-M23, M33 und der Armv8-M-Architektur. Für Neueinsteiger und wenig Geübte empfehlen wir die folgenden Trainings: Cortex®-Mx für Armv6-M/v7-M Architektur des Cortex®-M0/M0+/M3/M4 und M7 oder ARMv8-MS für Armv8-M Architektur des Cortex®-M23/M33 mit Security Extension und Embedded-C für Pointer, Strukturen, Funktionspointer, Speicherverwaltung in ANSI-C.

Live Online Training

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L-ARMV8MU

Präsenz-Training - Deutsch

Dauer

2 Tage

Live-Online - Englisch

Dauer

© MicroConsult Academy GmbH

Weitere Trainings auf www.microconsult.de. Änderungen vorbehalten.

Alle Preise sind Nettopreise pro Person zzgl. gesetzlicher USt.

Kontakt: info@microconsult.de, Tel. +49 (0)89 450617-71

2 Tage

Präsenz-Training - Englisch

Dauer

2 Tage

Cortex®-M23, M33: Armv8-M Architektur Training für Teilnehmer mit Kenntnissen der Vorgänger-Version - Live-Online-Training

Inhalt

TrustZone for Armv8-M

- Secure State Transitions
- Function calls from Secure state to Non-secure state
- Function returns from Non-secure state
- Praktische Übungen zum Entwickeln und Aufsetzen von gemischt Secure/Non-secure Projekten für den Cortex-M33

Cortex®-M (Armv8-M) Prozessor-Architektur

- Stack Limit Register
- Secure State, Security Transitions
- Banked Registers
- Cortex®-M Memory Map, System Control Block
- Praktische Übungen zu den neuen Stack Limit Registern

Unterschied zur Armv6-M und Armv7-M Prozessor-Architektur

Cortex®-M33, M23 Extended Instruction Set

- Branch and Control Flow Instructions with Security Transitions
- Security Instructions
- Assembler-Direktiven
- Praktische Übungen zur Erstellung kleiner Assembler-Routinen, zum Assembler-Debuggen und zur Code-Optimierung

Exception and Interrupt Handling

- Security Targeting
- Banked Exceptions
- Banked Vector Tables
- Tail Chaining with Security Transitions
- Interrupt Configuration and Status
- Secure Exception Priority Boosting
- Secure Faults
- Praktische Übungen zum SystemTick, Supervisor Call und PendSV im Kontext von RTOS-Anwendungen
- Praktische Übungen zu den Fault Handlern und Ausgabe von Status-Informationen

Memory Protection Unit MPU für Embedded-Systeme

- Armv8-M MPU
- Gegenüberstellung mit vorheriger Armv7-M MPU
- Praktische Übungen zur Programmierung der MPU
- Praktische Übungen zum dynamischen Umprogrammieren im Scheduler

Security Attribution Unit (SAU, IDAU)

- Überblick: Security und Implementation Defined Attribution Unit
- Attribution Attributes Secure, Non-secure, Non-secure Callable
- Praktische Übung zur Programmierung der Security Attribution Unit

Hardwarenahe C-Programmierung nach CMSIS

- CMSIS-Erweiterungen für Armv8-M

Übungen mit Keil µVision in Assembler und C

- Die Übungen werden mit Keil Studio (Visual Studio Code) durchgeführt. Keil uVision wird manchmal als Debugger verwendet