

Embedded-Multicore-Mikrocontroller in der Praxis - Live-Online-Training

Ziele - Ihr Nutzen

In diesem praxisorientierten Training lernen Sie die wesentlichen Mechanismen von Multicore-Mikrocontrollern und deren Leistungsfähigkeit kennen.

Neue Herausforderungen im Umgang mit Multicore meistern Sie effizient dank der Erfahrung, die Sie anhand zahlreicher Übungsbeispiele erwerben.

Mit dem neu erworbenen Know-how sind Sie gerüstet, selbständig eine zielgerichtete Multicore- μ C-Architekturauswahl durchzuführen und die Aufwände und Herausforderungen für ein Softwareprojekt abzuschätzen.

Probleme können Sie im Vorfeld identifizieren und wissen um entsprechende Lösungsmöglichkeiten.

Ihre Vorteile:

Praxisorientierter und zeitsparender Einstieg in die Gesamthematik

Praktische Tipps zur Performance-Optimierung und Software-Architektur

Zukunftssicherheit und Vermeidung von Design-Fehlern

Übungen als Download

Teilnehmer

Entwickler (Software, Hardware), Software-Architekten, Projektleiter und -manager, System-Architekten

Voraussetzungen

Mikrocontroller-Grundkenntnisse, Grundkenntnisse in C

Live Online Training

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L- μ CMULTI

Präsenz-Training - Deutsch

Termin	Dauer
01.07. – 03.07.2026	3 Tage

Live-Online - Englisch

Dauer
3 Tage

Präsenz-Training - Englisch

Termin **Dauer**
01.07. – 03.07.2026 3 Tage

Embedded-Multicore-Mikrocontroller in der Praxis - Live-Online-Training

Inhalt

Multicore-Mikrocontroller-Architektur

- Begriffsklärung von Multicore-Architekturen
- Homogene / heterogene Multicore-Architekturen mit shared Memory und/oder non-shared Memory
- Software-Aspekte für die Multicore-Verarbeitung
- Core-Interfaces und Memories: Core-local Cache und SPRAM (Level 1 Memory), Global/Shared SRAM (Level 2 Memory), Snoop-Logik (Cache-Kohärenz)
- Anforderungen an den Befehlsdurchsatz (MIPS)
- Core-Synchronisation
- Coprozessor-Funktionalität
- Neue Core-Bussysteme (Crossbar)
- Semaphore: Speicher-Zugriffssteuerung
- Speicherschutz (Zugriffschutz)
- Multicore Interrupt-Verarbeitung
- Multicore Start-up und Initialisierung: Boot-Prozess, Set-up von primären und sekundären CPUs
- Debug-Interface(s)

Multicore-Mikrocontroller Tool-Aspekte

- C/C++ Compiler: Erweiterungen für Multicore
- Lokatieren von Programm- und Data-Sections in spezifische Speicherbereiche, Steuerung der Zugriffsrechte für Global-/Extern-Definitionen
- Locator Safety Support: Variablen-Zugriffssteuerung für Multicore-Module

RTOS

- Multicore-Aspekte für RTOS-Software
- Scheduler: Softwareverteilungs- und -ausführungsstrategien
- Partitionierung
- Synchronisationskonzepte
- Kommunikationskonzepte
- Programmiermodelle und Multicore API: Kommunikation, Ressourcen-Management
- Beispiele für Multicore-unterstützende RTOS-Implementierungen

Multicore-Debugging und -Testaspekte

- Debugger-Erweiterungen für Multicore: Core-Synchronisation beim Debuggen, synchroner Start/Stop, Multicore Breakpoint Handling, Core-Context sensitive Darstellungen
- Performance- und Timing- Analyse, Analyse des Laufzeitverhaltens von Software (Profiling)
- Multicore und Trace-Aufzeichnungen

Safety

- Multicore in den Normen
- Hardware Safety Measures
- Safety Management Unit SMU
- Bus Error Detection und Protection Mechanismen
- Safety Core (Checker Core, Lockstep Core)
- Safety on-Chip Testfeatures

Übungen: Werden auf Evaluierungsboards basierend auf Aurix-Mikrocontrollern durchgeführt

- Multicore-Startverhalten
- Speicher-Allokation und -Partitionierung
- Dekomposition von bestehenden Singlecore-Applikationen
- Portierung auf Multicore
- Synchronisation/Kommunikation
- Schutzmechanismen
- Performance-Messungen

HINWEIS: Die Kursunterlagen sind auf Englisch