

Architectural Design of Real-Time Software

HILF! GmbH Microcomputer-Consulting, Bajuwarenring 17, 82041 Oberhaching/München
Tel.: +49 (89) 61 37 90 - 0, Fax: +49 (89) 62 52 14 5

Web: www.hilf.de
Email: info@hilf.de

Target group

This course is intended for practicing real-time and embedded systems software system architects, project managers and technical consultants who have responsibility for designing, structuring and implementing the software for real-time and embedded computer systems using an RTOS.

Prerequisites

Course participants are expected to have thorough familiarity with at least one RTOS.

Course description

This course examines the activities of high-level design of real-time and embedded systems software that's to be developed using a real-time operating system (RTOS).

The class begins with a quick examination of some fundamental issues in real-time multitasking embedded application software design and development, and briefly reviews several modern techniques for real-time and embedded software requirements specification. It then quickly focuses on how to structure a software system that must execute within strict deadline and resource limits. Emphasis is placed on multitasking and timing behaviors, rather than object orientation.

The class continues with a detailed examination of a broad spectrum of intertask communication and synchronization options including mutexes of several varieties. 'Liveness' issues such as deadly embrace, lockout, memory starvation and CPU starvation will be discussed in detail. Students will learn how to correctly configure queue lengths, and examine design dangers such as interrupt overflow and memory starvation. The next major subject area of the class is the evaluation of timing performance and quality of a real-time or embedded software design.

This course is not a general course about software design theory, but rather it is highly focused on the design of deeply-embedded, time-constrained, resource-constrained multitasking software that will run under the control of a modern RTOS.

Course topics

- Day 1 - Morning: Definitions / Real-Time Requirements
 - Embedded / Real-Time Systems: Definitions and Issues
 - Requirements Analysis: Context Diagram and Usage Scenarios
 - Quick Overview of Entity-Relationship Diagrams, State Transition Diagrams and Statecharts
 - What You Get in an RTOS, and What You Don't Get
- Day 1 - Afternoon: Identifying Tasks and Objects
 - High-Level Software Architecture: Concurrency
 - The Goma Guidelines for Decomposition into Tasks
 - Exercise: Heart Rhythm Monitor - Design Partitioning
 - Object Orientation for Embedded/Real-Time Software?
- Day 2 - Morning: 13 Methods for Intertask Communication
 - The Correctness Killers: Deadlock, Lockout, Starvation
 - Software Architecture: Intertask Data Communication
 - Software Architecture: Intertask Synchronization
- Day 2 - Afternoon: Fine-Tuning of Intertask Communication
 - Modeling of Message Queues
 - Exercise: Queuing Calculations for Medical Device
 - Mutexes: Priority Inheritance and Priority Ceiling
- Day 3 - Morning: A Plethora of Design Examples
 - "Standard Circuits" for Real-Time and Multitasking Software Design
 - Design Patterns for Embedded Software
 - Extended Example: Aircraft Fly-by-Wire
- Day 3 - Afternoon: Evaluating the Quality of a Multitasking Software Design
 - Evaluating Real-Time and Multitasking Software Designs
 - Performance Analysis: Individual Scenarios
 - Performance Analysis: Task Scheduling Theory and Calculations
 - Exercise: Medical System Performance Calculations