

Design Patterns for Embedded and Real Time Systems

Course Number 3867 – 24 Hours

Overview

The design of Real Time and embedded systems must be powerful, efficient, and highly reliable. The constraints of processor and memory resources add to this challenge. Sophisticated developers rely on design patterns for building fail-safe RTE systems.

Key topics:

- Design Principles in RTE systems
- Object Oriented Design
- Coordinating and organizing system components and subsystems
- Managing memory and resources
- Building safe and reliable architectures
- Mapping subsystem and component architectures to underlying hardware

Prerequisites

Delegates should have a working knowledge of C/C++, and some knowledge of Embedded/Real Time programming

Course Contents

Overview

- Real time systems design
- Design patterns overview
- Patterns examples

Languages

- C pitfalls
- C++ in embedded and real time systems
- Embedded C++ specifications

Implementing Object Oriented Concepts with C

- Classes
- Inheritance
- Polymorphism

Memory Management – Advance techniques

- New and delete operators
- Common pitfalls
- Static allocation
- Pool allocation
- Smart pointer and proxy pattern
- Garbage collection pattern
- Shared memory pattern

Object Patterns

- Manager pattern
- Resource Management Patterns
- Message Handler
- Observers and Publish – Subscribe Patterns

State Machines Patterns

- State Pattern
- Serial and parallel wait state Patterns
- Hierarchical Pattern

Concurrency Patterns

- Concurrency pattern
- Message Queue pattern
- Interrupt pattern
- Cyclic executive
- Static and dynamic priorities
- Priority inversion
- Simultaneous locking
- Concurrency problems
- Parallel programming
- Case study: OpenMP

Hardware Interface Patterns

- Serial Interface pattern
- Hardware devices pattern
- Synchronizer pattern
- Timer management patterns

Protocols and Architectures Patterns

- Transmit and Receive handlers
- Stack and layers pattern
- Layered patterns
- Microkernel pattern
- Virtual machines

Appendixes

- Effective STL
- UML
- Case study: Embedded system design with UML