FEABHAS

Robust Software Development

Course category	Software engineering
Training area	Quality
Course code	RBC-201
Duration	2 days
Additional information	For price: Contact Feabhas (available for on-site delivery only)

This course targets the construction of robust, safe embedded systems using C. The focus is on developing robust, safe software in a concurrent, multi-threaded environment.

For each topic, the issues and consequences of a design or implementation will be explored, allowing the attendee to make reasoned choices when they are back in their work environment.

Pre-requisites:

A good knowledge of C.

Who should attend:

This course is aimed at experienced C developers.

Duration:

- Two days

Course materials:

- Delegate handbook

Hardware Manipulation

- Pointer syntax and semantics (review)
- Hardware access via pointers
- Bit-field structures
- Issues with bit-field structures (interpretation, padding, etc)
- Hardware access using structures
- Issues with structures for hardware access

State-machine Fundamentals

- Reactive objects and modal behaviour
- State transitions
- Doing work in the state machine

Function Pointers

- Function pointer syntax
- Polymorphic code
- Issues with polymorphic code
- State machine implementation

Memory Management

- Problems with dynamic memory (overview)
- Fixed block allocation schemes
- Tuning issues with fixed-block allocators

Numerical Issues

- Problems with fixed-point numbers
- Problems with floating-point numbers
- The MISRA-C Essential Types model

Exception Handling

- What is an exception?
- The 'four layers of exception' – Ignore, Handle, Degrade and Halt
- Defining an exception strategy

Interrupts

- The need for interrupts
- Understanding priority
- Interrupts as pseudo-concurrency
- Internal and external interrupt controllers
- Interrupt service routine programming

Multi-tasking

- Tasks, threads, processes and platforms
- Context switching
- Scheduling methods
- Task creation

Mutual Exclusion

- Race conditions and critical sections
- Simulating critical sections
- Priority inversion
- Recursive deadlock
- Cyclic dependency deadlock

Task Synchronisation

- The problem with busy-waiting
- Condition variables
- The Monitor pattern

Feabhas Ltd - UK Office 15-17 Lotmead Business Park, Wanborough, Swindon, SN4 0UY, UK •

+44 (0) 1793 792909 • info@feabhas.com • www.feabhas.com