



TRAINING IN REAL-TIME
EMBEDDED DEVELOPMENT

Course C-501: C for Real-Time Developers

Course Description:

Learning the C programming language is one thing, but learning how to use it effectively and appropriately in a real-time embedded environment is another. For many programmers, these skills are learnt the hard way, through trial and error on the job. This course addresses this issue by not only teaching the C programming language, but through emphasising the pragmatic use of C in a real-time environment. This is achieved through both lecture and direct hands-on experience.

Approximately 50% of the course consists of writing code for a real target. During the week attendees will build up a complete solution to a case study that exercises all significant parts of the course.

Overview:

This is an intensive five day course covering C in the context of real-time embedded application development.

Course Objectives:

- To provide an understanding of the essentials of the C programming language.
- To give you practical experience of writing C for real-time and embedded systems.
- To demonstrate the traps and pitfalls of the language when used in a real-time system.
- To give you the confidence to apply these new concepts to your next real-time project.

Delegates will learn:

- The core C syntax and semantics
- How to access hardware and program interrupts in the language
- About memory and performance issues associated with C
- How real-time operating systems (RTOS) affect the use of the language

Pre-requisites:

- Experience of programming, e.g. assembly language
- Prior knowledge of C is useful but not essential

Who should attend:

This course is designed for engineers who are embarking on a project using C for the first time. It is also suitable for those needing to support a customer using C, and those requiring to be brought up to date since initially being taught at University.

Duration:

Five days

Course Materials:

- Delegate handbook

Related Courses:

- RTOS-201 Fundamentals of Real-Time Operating Systems
- AC-401 Advanced C Programming
- MC-101 Guidelines for Writing High Integrity Software in C
- 00-503 Real-Time Software Design with UML2.0

Course Workshop:

This course utilises target hardware during the embedded programming practical exercises. The target board is an ARM7-based microcontroller (NXP LPC2129) connected to an application board. Programming the target microcontroller to control the application board gives attendees a real-sense of embedded application development.

Course Outline:

Introduction to Real-Time Systems

- Characteristics and demands of real-time software development

Introduction to C Programming

- Structure of a C program
- The C compilation process

Types and Operators

- C base types
- Precedence & associativity
- Arithmetic operations

Control Flow

- Logical expressions and operations
- Decision Making
- Loops

Pointers

- Pointer basics

Bit Manipulation & Hardware Access

- Accessing hardware with pointers
- Manipulating information at the bit level
- General Purpose IO (GPIO)

Functions

- The Function as a logical program unit
- How parameters are passed
- Memory segments

Arrays, Pointers and Strings

- Arrays as circular buffers
- Relationship between pointers & arrays
- Pointer arithmetic
- C string handling

Device Interaction &

Synchronization

- Polling devices
- Serial peripheral programming

Structures and Unions

- Structures;
- Big & Little Endian
- Unions
- Using structures to access I/O devices
- Bit-field structures

Interrupts

- Internal & External interrupt control
- Nested Interrupts
- Priorities & Masks
- Software Interrupts

Dynamic Memory Management

- Malloc and free
- Linked list
- Issues (leaks, fragmentation, etc.)

Program Structure

- Definitions and declarations;
- Header files
- Scope and lifetime

The C Pre-Processor

- Macros
- Conditional Compilation

Program Design Concepts

- Why modular design?
- Coupling and cohesion
- Testing the program

Concurrency

- Scheduling strategies
- Context switching
- Function pointers
- Task synchronisation and communication
- MMU and memory protection
- Concurrency and libraries

The Standard C Library

Target Specific Considerations

- Language features affecting portability
- Non-standard C language features
- Assembly language interfacing
- What happens at power-on?
- Initialising the C runtime

Course Summary

FEABHAS

Feabhas Ltd

5, Lowesden Works
Lambourn Woodlands
Hungerford, Berkshire
RG17 7RY, UK

Tel: +44 (0) 1488 73050

Fax: +44 (0) 1488 73051

Email:

info@feabhas.com

Web:

www.feabhas.com