

## Advanced C Programming

---

<b>Course category</b>	C Training Courses
<b>Training area</b>	Programming Languages
<b>Course code</b>	AC-401
<b>Duration</b>	4 days
<b>Price exc VAT</b>	£2400.00
<b>Additional information</b>	Public courses delivered remotely over Zoom, using QEMU to emulate the hardware

Many engineers work with the C programming language every day for real-time embedded development. However, some hit a plateau and never feel comfortable with certain aspects of the language.

This course aims to take individuals to the next level and give them a fuller appreciation of the more advanced aspects of the language. Due to the requirements for programming real-time embedded systems, this course goes beyond just addressing the language issues and explores compile, link and run-time issues.

In addition, it covers hardware access, interrupts and multi-tasking areas specific to the C programming language.

50% of the course is spent writing code for a real target.

### Course objectives:

- To become comfortable with the advanced aspects of the C programming language.
- To gain an in-depth knowledge on what is happening at compile, link and runtime on a target processor.
- To introduce good quality and style for real-time embedded programming.
- To gain hands-on experience of programming up interrupts and real-time operating systems (RTOS).

### Delegates will learn:

- Advanced aspects of the C programming language (C99)
- The traps and pitfalls of the language (e.g. structure packing, dynamic memory, etc.)
- Compile, link, and run-time memory models
- MISRA-C ([www.misra.org.uk](http://www.misra.org.uk)) guidelines
- What parts of C should and should not be used in real-time embedded systems programming

**Pre-requisites:**

- A good grasp of the fundamentals of C

or

- Attendance of Feabhas course C-501: C for Real-Time Developers.

**Who should attend:**

This course is designed for engineers who want to improve their grounding and understanding of the C programming language. It is specifically aimed at issues relevant to real-time embedded software engineers.

**Duration:**

- Four days

**Course materials:**

- Delegate handbook

**Course workshop:**

The course workshop uses the GCC-ARM compiler, Eclipse IDE and the STM32F407VG ARM Cortex-M4 microcontroller as a target for the hands-on sessions. Exercises are designed to stretch attendees' abilities and explore subtleties of the language.

A good working knowledge of the C programming language is essential as some of the exercises involve extensive programming.

**Day 1****Program objects**

- Declarations and definitions
- Expressions
- Scope and lifetime

**The build process**

- compilation, linking, memory sections, libraries

## **System startup & runtime**

- What happens before main
- What is happening at runtime

## **Pointers**

## **Function calling**

- Parameter passing mechanism
- Pass-by-value vs Pass-by-reference
- Returning data objects

## **Day 2**

## **Hardware manipulation**

- Register access
- The volatile keyword

## **Arrays**

- Initialisation
- Relationship to pointers

## **State machine basics**

- Mealy & Moore machines

## **Function Pointers**

- Callbacks
- State machine

## **Day 3**

## **Dynamic Memory**

- Malloc, calloc and realloc
- Variable sized structures
- Dangers

## **Structures, Unions and Bitfields**

- Definition & use
- Memory layout

## **Numerical issues**

- MISRA essential types
- Floating point issues
- Fixed point types

## **Day 4**

### **Interrupts**

- Hardware interrupt models
- Software interrupts

### **Multi-Tasking (Multi-threading)**

- Terminology
- Advantages and disadvantages
- Mutual exclusion
- Intertask communication & synchronisation