



## Real-Time

### Programmation et conception temps réel

Creating systems that work in real-time is a specific challenge. That's why **ac6-training** provides a range of courses to explain you all the specific techniques and tools to use in this context.

You can see detailed course descriptions of the various trainings by using the above navigation bar. You can also click on course identifiers in the following course briefs hereafter.

#### Main Courses

**MC4 - Multi-Core Programming with OSEK/VDX and AutoSAR** Programming real-time and multi-core systems, avoiding common pitfalls

Electronics Control Units used in Automotive systems are more and more powerful and are now using multicore processors, causing specific problems to integrate applications that may not have been designed with multicore in mind. This training helps you master multitask and real-time programming of multi-core processors in the Automotive sector, understanding how to effectively solve problems using the primitives provided by the underlying Operating System.

**RT1 - Real Time and Multi-Core programming** Programming Linux real-time and multi-core systems, avoiding common pitfalls

Real-time and embedded code, especially targeting multicore processors, cannot be effectively tested; it must be validated before coding. This training help you master multitask and real-time programming of multi-core processors, understanding how to effectively solve problems using the primitives provided by the underlying Operating System.

**RT3 - FreeRTOS Real Time Programming** Real-time programming applied to the FreeRTOS operating system

**RT5 - Zephyr Real Time Programming** Real-time programming applied to the Zephyr operating system

**RT6 - Real Time Programming with – Azure RTOS ThreadX** Real-time programming applied to the Azure RTOS (ThreadX) operating system

#### Additional Courses

**C7 - UML Real-Time** UML for embedded and real-time systems

**C8 - Critical Systems Safety** Embedded systems are more and more critical and subject to safety constraints. This training introduces the main concepts and standards applicable to safety-critical systems.

**C9 - Software Architecture with UML** Embedded systems are increasingly complex and therefore can no longer be designed using traditional methods.

**D4 - Real-time Linux** Real-time Linux with RT-Preempt patch and Xenomai, to measure real-time performance

**IOT1 - Internet of Things (IoT) on Microcontrollers** Building low-power IoT devices using standard microcontrollers

**L5 - Real time Java (TM)** Programmation temps réel en Java

**STG - STM32 + FreeRTOS + LwIP** This course covers the STM32 ARM-based MCU family, the FreeRTOS operating system and the LwIP network stack.