# ac6

## 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.

### 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 mutitask 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 targetting multicore processors, cannot be effectively tested; it must be validated before coding. This training help you master mutitask 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 Eclipse ThreadX** Real-time programming applied to ThreadX (previously Azure RTOS)

### 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 more be directly designed using existing schemes. One need to first create a detailed architecture to control and plan their development and integration appropriately. This course will help address these phases efficiently and avoid common pitfalls; it will explain you why Software Architecture is needed and how architecture processes can be implemented in an enterprise environment.

**D4 - Real-time Linux** Real-time Linux with RT-Preempt patch and Xenomai This course presents the various solutions for a real-time Linux and the tools to measure real-time performances

**IOT1 - Internet of Things (IOT) on Microcontrollers** Building low-power IOT devices using standard microcontrollers Th is course introduce the IoT ecosystem, describe the most used IoT Edge to Cloud Protocols (MQTT, MQTT-SN and CoAP), explore particularly heinous IoT focused attacks and security provisions at each level of stack (physical devices, communication systems and networks) . This course explains how to configure the LwIP (with MQTT), FreeRTOS and MbedTLS for a microcontroller-based IoT application; it requires previous knowledge of FreeRTOS.

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

**STG - STM32 + FreeRTOS + LwIP** This course covers the STM32 ARM-based MCU family, the FreeRTOS Real Time OS, the LWIP TCP/IP Stack and/or the EmWin GUI Stack

**TI3 - Cortex M4 Texas Instruments Implementation and Ti-RTOS**This course describes the Texas Instruments ARM Cortex M4F implementation and TI-RTOS real-time programming