

# Real-time systems



## Component

École Nationale  
Supérieure  
d'Électrotechnique  
d'Électronique  
d'Informatique  
d'Hydraulique  
et des  
Télécommunications

## In brief

- > **Amety's Code:** N8EE17C
- > **Open to exchange students:** Yes

## Presentation

---

### Objectives

Critical computer systems are omnipresent in our daily lives. They control our cars, fly our planes, distribute our electricity, etc. These systems are considered critical because their failure can have dramatic consequences, whether financial, material, or human. Critical systems must therefore offer strong guarantees regarding the correctness of their execution and meet certain responsiveness requirements.

We will see how a real-time operating system can help provide these guarantees. We will present rigorous analysis methods for ensuring system responsiveness. We will also show how these results can be used to help optimize system design.

---

### Description

In the first part, the module will focus on traditional methods for sharing a computer between several tasks. This issue will be addressed through the creation of a simple task scheduler that can be implemented on an Arduino-type microcontroller. We will thus present the specific features of a real-time operating system designed for critical embedded systems, and in particular real-time scheduling algorithms. These algorithms enable the system's schedulability to be analyzed, which makes it possible to formally prove the correctness of the system's responsiveness.

In the second part, we will present the OSEK/VDX standard, which is widely used in real-time operating systems in the automotive sector and forms the basis of the AUTOSAR standard. The concepts will be applied in a design office in an embedded context with the Trampoline RTOS real-time operating system.

---

## Pre-requisites

A good knowledge of algorithms, C programming, and operating system principles, particularly Linux, is required.