

**Module Description, available in: EN**

## *Embedded Real-time Software*

**General Information****Number of ECTS Credits**

3

**Module code**

TSM\_EmbReal

**Valid for academic year**

2025-26

**Last modification**

2024-10-17

**Coordinator of the module**

Simon Künzli (ZHAW, simon.kuenzli@zhaw.ch)

**Explanations regarding the language definitions for each location:**

- Instruction is given in the language defined below for each location/each time the module is held.
- Documentation is available in the languages defined below. Where documents are in several languages, the percentage distribution is shown (100% = all the documentation).
- The examination is available 100% in the languages shown for each location/each time it is held.

	Lausanne			Lugano	Zurich		
<b>Instruction</b>					X E 100%		
<b>Documentation</b>					X E 100%		
<b>Examination</b>					X E 100%		

**Module Category**

TSM Technical scientific module

**Lessons**

2 lecture periods and 1 tutorial period per week

**Entry level competences****Prerequisites, previous knowledge**

- Knowledge of C programming language and object-oriented programming in a programming language
- Good knowledge of computer and microprocessor architectures
- Experience in implementing a project on a microcontroller in C
- Basic knowledge of Operating Systems necessary, basic knowledge of RTOS recommended
- Basic knowledge of concurrent programming

## Brief course description of module objectives and content

Embedded Systems, although they are not visible, have become integral parts of this world. Embedded Systems essentially consist of two components: hardware and software. In contrast to information systems e.g. in the banking world, hardware is more application specific. Due to this fact, the software that interacts directly with the hardware is more specific as well. Real-time and concurrency are important issues in Embedded System development, which come on top of the generally valid requirements for correctness and reliability.

The module teaches methods to develop Real-Time Embedded System Software and deals with the following complementary aspects:

- Real-Time Operating Systems, Applications and Modelling
- Modern C++ for microcontrollers, focusing on programming close to hardware with and without dynamic memory allocation
- Software concepts for asymmetric multiprocessor systems

## Aims, content, methods

### Learning objectives and competencies to be acquired

Based on requirements, the students will be able to apply the optimal method to develop and verify an Embedded System,

- on the boundary between hard- and software using modern C++ and RTOS features
- on application layer using modeling methods.

### Module content with weighting of different components

The module provides insights at real-time embedded systems from various perspectives, consisting of three major parts.

In the first part, we discuss SW modelling and implementation aspects using real-time operating systems

- Introduction to RTOS
- Task Models and Real-Time Scheduling
- Concurrency
- Modeling & Code Generation
- Testing & Debugging

In part II, we focus on C++ for embedded systems with focus on the use on microcontrollers.

- Using C++: showing the huge advantages of C++ for Embedded Systems
- Point out where C++ uses dynamic memory allocation and how to deal with it on microcontrollers.

In the last part, the focus is on software development for asymmetric multi-core embedded systems:

- Inter-Processor communication
- Multi-core programming

### Teaching and learning methods

- Ex-cathedra teaching
- Exercises
- Self-study (study of papers, case studies)
- practical exercises: programming embedded software on embedded systems

### Literature

## Assessment

### Additional performance assessment during the semester

The module does not contain an additional performance assessment during the semester

### Basic principle for exams

**As a rule, all standard final exams are conducted in written form. For resit exams, lecturers will communicate the exam format (written/oral) together with the exam schedule.**

### Standard final exam for a module and written resit exam

**Kind of exam**

Written exam

**Duration of exam**

120 minutes

**Permissible aids**

*Aids permitted as specified below:*

**Permissible electronic aids**

- open book, all materials including PC, laptop, tablets etc. allowed during the exam session
- any telecommunication and the respective programs/apps for communications must be turned off (incl. mobile phones)
- no active help from third parties, i.e. no active participation in web forums, discussion groups, or similar means of information exchange

**Other permissible aids**

No other aids permitted

**Exception: In case of an electronic Moodle exam, adjustments to the permissible aids may occur. Lecturers will announce the final permissible aids prior to the exam session.**

**Special case: Resit exam as oral exam****Kind of exam**

Oral exam

**Duration of exam**

30 minutes

**Permissible aids**

No aids permitted