Module Description, available in: EN

Embedded Real-time Software

General Information

Number of ECTS Credits

3

Module code

TSM_EmbReal

Valid for academic year

2026-27

Last modification

2025-10-17

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 specified for each location and module execution.
- Documentation is available in the language(s) listed for each location and module execution. If the documentation is in multiple languages, the percentage distributed is indicated (100% = all documentation provided).
- The examination, including both questions and answers, is provided entirely (100%) in the language(s) specified for each location and module execution. The exams are on-site.

	Lausanne			Lugano	Zurich		
Instruction					**X** E 100%		
Documentation					**X** E 100%		
Examination					**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

No electronic aids permitted.

Other permissible aids

Up to 4 A4 pages of a self-written summary are 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