Module Description, available in: EN

Embedded Real-time Software

General Information

Number of ECTS Credits

3

Module code

TSM_EmbReal

Valid for academic year

2020-2021

Last modification

2019-10-11

Responsible of module

Hans Buchmann (FHNW, hans.buchmann@fhnw.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.

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

- Programming language C++/C
- Computer architectures
- Fundamentals of Operating Systems

Brief course description of module objectives and content

Embedded Systems, although they are not visible, they have become integral parts of this world. Embedded Systems essentially consist of two components, hardware and software. In contrast to information systems 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 Embedded System Software and deals with the following two complementary aspects:

- Embedded Programming, Programming close to hardware
- Abstract Modeling Concepts.

Both parts are based on Object-Oriented Concepts.

Aims, content, methods

Learning objectives and acquired competencies

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++,
- on application layer using modeling methods.

Contents of module with emphasis on teaching content

In the first part, the focus is on Near-Hardware-Programming. We use a typical (small) System on Chip (SoC) equipped with a RISC V.

The programming language is C++, the programming environment is Linux.

- Using C++: showing the huge advantages of C++ for a small SoC
- ISA Instruction Set Architecture
- Hardware-Access
- Concurrency
    - for a SoC
    - for a Linux based System

In the second part, the focus is on modeling, a model driven approach: from requirements, over modeling to the running system

- Introduction
    - Development Process
    - Generic Software-Architecture
- Modeling functional requirements
    - System of cooperating state machines
    - CIRO (Communicating Interacting Reactive Objects)
- Modeling connection software
    - Connection between hardware and reactive system
- Code Generation
    - Generated Code
    - Strategies
    - Tools?
- Testing executable Models
- Real-Time Scheduling
    - Multi-Tasking
    - Distribution
    - Task and Event Scheduling
- Exercises and laboratories using concrete tool-chain and microcontroller

Teaching and learning methods

- Ex-cathedra teaching
- Exercises
- Self-study (study of papers, case studies)

Literature

Assessment

Certification requirements

Module does not use certification requirements

Basic principle for exams

As a rule, all the standard final exams for modules and also all repetition exams are to be in written form

Standard final exam for a module and written repetition exam

Kind of exam

written

Duration of exam

120 minutes

Permissible aids

No aids permitted

Special case: Repetition exam as oral exam

Kind of exam

oral

Duration of exam

30 minutes

Permissible aids

No aids permitted