

Module Description, available in: EN, FR

Embedded Real-time Software

General Information

Number of ECTS Credits

3

Module code

TSM_EmbReal

Valid for academic year

2019-20

Last modification

2018-10-29

Coordinator of the 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 F 100%		X E 100%
Documentation			X E 100%	X E 100%
Examination		X F 100%	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 parallelism 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 competencies to be acquired

Based on requirements, the students will be able to apply the optimal method to develop and verify an Embedded System, covering software on the boundary between hard- and software using C++, as well as the application layer using modeling methods.

Module content with weighting of different components

In the first part, the focus is on Near-Hardware-Programming, we use a typical (small) system on chip. The programming language is C++, the programming environment is Linux.

- Hardware-Access
- Interrupts
- Parallelism

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

- Software-Architecture
- Modeling extended Unified Modeling Language (UML)
- Testing of executable Models
- Real-Time Scheduling

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 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

No aids permitted

Special case: Resit exam as oral exam

Kind of exam

Oral exam

Duration of exam

30 minutes

Permissible aids

No aids permitted

Description du module, disponible en: EN, FR

Informatique temps-réel embarquée

Informations générales

Nombre de crédits ECTS

3

Code du module

TSM_EmbReal

Valable pour l'année académique

2019-20

Dernière modification

2018-10-29

Coordinateur/coordinatrice du module

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

Explications concernant les langues d'enseignement par site :

- Les cours se dérouleront dans la langue définie ci-dessous par lieu/exécution.
- Les documents sont disponibles dans les langues définies ci-dessous. Pour le multilinguisme, voir la répartition en pourcentage (100% = documents complets)
- L'examen est disponible à 100% dans chaque langue sélectionnée pour chaque lieu/exécution.

	Berne	Lausanne	Lugano	Zurich
Leçons		X F 100%		X E 100%
Documentation			X E 100%	X E 100%
Examen		X F 100%	X E 100%	X E 100%

Catégorie de module

TSM approfondissement technico-scientifique

Leçons

2 leçons et 1 leçon de pratique par semaine

Compétences préalables

Connaissances préalables, compétences initiales

- Langage de programmation C/C++
- Architectures d'ordinateur
- Notions fondamentales sur les systèmes opérationnels

Brève description du contenu et des objectifs

Les systèmes embarqués, bien que n'étant pas visibles, sont devenus des parties intégrantes de notre monde. Les systèmes embarqués sont pour l'essentiel constitués de deux composants, matériel et logiciel. Contrairement aux systèmes informatiques du monde bancaire, le matériel est plus spécifique aux applications. Par conséquent, le logiciel interagissant avec le matériel est lui aussi plus spécifique.

Temps réel et parallélisme sont d'importants critères dans le développement des systèmes embarqués, et sont en tête des exigences usuelles de correction et de fiabilité.

Le module enseigne des méthodes de développement des systèmes embarqués et traite des deux aspects complémentaires suivants:

- Programmation embarquée (programmation proche du matériel)
- Concepts de modélisation abstraite. Ces deux parties se basent sur des concepts à orientation objet

Objectifs, contenus, méthodes

Objectifs d'apprentissage, compétences à acquérir

Suivant les besoins, les étudiants seront à même d'appliquer la méthode optimale afin de développer et de vérifier un système embarqué couvrant le logiciel de la frontière entre matériel et logiciel au moyen du langage de programmation C++, de même que la couche applicative au moyen de méthodes de modélisation.

Contenu des modules avec pondération du contenu des cours

La première partie met l'accent sur la programmation proche du matériel, nous avons recours à un (petit) système sur puce électronique. Le langage de programmation est C++, l'environnement de programmation est Linux.

- Accès au matériel
- Interrupts
- Parallélisme

La seconde partie est centrée sur la modélisation, une approche à orientation modèle: des exigences à la modélisation et à l'exploitation du système.

- Architecture logicielle
- Modélisation du langage étendu Unified Modeling Language (UML)
- Test de modèles exécutables
- Planification en temps réel

Méthodes d'enseignement et d'apprentissage

- Enseignement magistral
- Exercices
- Etude personnelle (études de documents, études de cas)

Bibliographie

Evaluation

Conditions d'admission

Le module n'utilise pas de conditions d'admission.

Principe pour les examens

En règle générale, tous les examens réguliers de fin de module se déroulent sous forme écrite. Concernant les examens de répétition, leur format (écrit ou oral) sera communiqué par l'enseignant-e en même temps que le calendrier des examens.

Examen de fin de module régulier et examen écrit de répétition

Type de l'examen

Examen écrit

Durée de l'examen

120 minutes

Aides autorisées

Sans aides

Cas spécial: examen de répétition oral

Type de l'examen

Examen oral

Durée de l'examen

30 minutes

Aides autorisées

Sans aides