# Table of Contents

Mathematics for Signals & Controls (MOD1-01) .................................................. 7
Distributed and Parallel Systems (MOD1-02) ....................................................... 9
Embedded Software Engineering (MOD1-03) ..................................................... 11
Requirements Engineering (MOD1-04) ............................................................... 14
Scientific & Transversal Skills (MOD1-05) ......................................................... 16
Mechatronic Systems Engineering (MOD2-01) ................................................. 18
Microelectronics & HW/SW-Co-Design (MOD2-02) ....................................... 20
R&D Project Management (MOD2-03) ................................................................. 22
Signals & Control Systems 1 (MOD2-04) ............................................................. 25
Research Project (Thesis) (MOD3-03) ................................................................. 27
Master Thesis + Colloquium (P) ........................................................................ 29
Applied Embedded Systems (MOD-E01) .......................................................... 32
Smart Home & Smart Building & Smart City (MOD-E02) ................................ 34
SW Architectures for Embedded and Mechatronic Systems (MOD-E03) .......... 36
Signals and Systems for Automated Driving (MOD-E04) ............................... 38
IoT & Edge Computing (MOD-E05) .................................................................. 40
Computer Vision (MOD-E06) ............................................................................. 42
Signals & Control Systems 2 (MOD-E07) .......................................................... 44
Formal Methods (MOD-E08) ............................................................................... 46
System on Chip Design (MOD-E09) .................................................................. 48
Automotive Systems (MOD-E10) ....................................................................... 50
Hardware Project (MOD-E11) ............................................................................ 52
Model Based Systems Engineering (MOD-E12) ................................................ 54
Software for Robots (MOD-E13) ........................................................................ 56
Embedded Systems Hardware Design and Rapid Prototyping (MOD-E14) ....... 58
Trends in Embedded and Mechatronic Systems (MOD-E15) ............................ 60
# Study Programme overview

## 1st Semester (Winter Semester)

<table>
<thead>
<tr>
<th>Module</th>
<th>Exam</th>
<th>ModNr/ PNr</th>
<th>Workload</th>
<th>Contact Time</th>
<th>Self Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematics for Signals &amp; Controls</td>
<td>MOD1-01</td>
<td>10110/11</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Distributed and Parallel Systems</td>
<td>MOD1-02</td>
<td>10120/21</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Embedded Software Engineering</td>
<td>MOD1-03</td>
<td>10130/31</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Requirements Engineering</td>
<td>MOD1-04</td>
<td>10140/41</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Scientific &amp; Transversal Skills</td>
<td>MOD1-05</td>
<td>10160/61</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td>5</td>
<td>20</td>
<td>300</td>
</tr>
</tbody>
</table>

## 2nd Semester (Summer Semester)

<table>
<thead>
<tr>
<th>Module</th>
<th>Exam</th>
<th>ModNr/ PNr</th>
<th>Workload</th>
<th>Contact Time</th>
<th>Self Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechatronic Systems Engineering</td>
<td>MOD2-01</td>
<td>10210/11</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Microelectronics &amp; HW/SW Co-Design</td>
<td>MOD2-02</td>
<td>10220/21</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>R&amp;D Project Management</td>
<td>MOD2-03</td>
<td>10230/31</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Signals and Control Systems 1</td>
<td>MOD2-04</td>
<td>10240/41</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Elective 1 *</td>
<td>MOD2-05</td>
<td>10250</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td>5</td>
<td>20</td>
<td>300</td>
</tr>
</tbody>
</table>

## 3rd Semester (Winter Semester)

<table>
<thead>
<tr>
<th>Module</th>
<th>Exam</th>
<th>ModNr/ PNr</th>
<th>Workload</th>
<th>Contact Time</th>
<th>Self Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elective 2 *</td>
<td>MOD3-01</td>
<td>10310</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Elective 3 *</td>
<td>MOD3-02</td>
<td>10320</td>
<td>4</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Research Project (Thesis)</td>
<td>MOD3-03</td>
<td>10330/31</td>
<td>0</td>
<td>0</td>
<td>540</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td>3</td>
<td>8</td>
<td>120</td>
</tr>
</tbody>
</table>

## 4th Semester (Summer Semester)

<table>
<thead>
<tr>
<th>Module</th>
<th>Exam</th>
<th>PNr</th>
<th>Workload</th>
<th>Contact Time</th>
<th>Self Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Thesis and Colloquium</td>
<td>P</td>
<td>103</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>900</td>
</tr>
</tbody>
</table>

1 SWS= weekly hours per semester

*cf. Catalogue of Elective Modules*
# Catalogue of Elective Modules

## Catalogue of Elective Modules (Electives 1, 2 and 3)*

<table>
<thead>
<tr>
<th>Modul</th>
<th>Module nr.</th>
<th>Exam nr.</th>
<th>Workload</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applied Embedded Systems</td>
<td>MOD-E01</td>
<td>10401</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Smart Home &amp; Smart Building &amp; Smart City</td>
<td>MOD-E02</td>
<td>10414</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>SW Architectures for Embedded and Mechatronic Systems</td>
<td>MOD-E03</td>
<td>10403</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Signals and Systems for Automated Driving ***</td>
<td>MOD-E04</td>
<td>10404</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>IoT &amp; Edge Computing</td>
<td>MOD-E05</td>
<td>10413</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Computer Vision ***</td>
<td>MOD-E06</td>
<td>10406</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Signals &amp; Control Systems 2 ***</td>
<td>MOD-E07</td>
<td>10407</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Formal Methods</td>
<td>MOD-E08</td>
<td>10408</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>System on Chip Design</td>
<td>MOD-E09</td>
<td>10409</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Automotive Systems</td>
<td>MOD-E10</td>
<td>10410</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Hardware Project</td>
<td>MOD-E11</td>
<td>10412</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Model Based Systems Engineering</td>
<td>MOD-E12</td>
<td>10415</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Software for Robots</td>
<td>MOD-E13</td>
<td>10416</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Embedded Systems Hardware Design and Rapid Prototyping</td>
<td>MOD-E14</td>
<td>10417</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Trends in Embedded and Mechatronic Systems</td>
<td>MOD-E15</td>
<td>10418</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>Research Seminar</td>
<td>S</td>
<td>10411</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Module(s) from cooperating institutions    |            |          |          |      |      | 10421 |
| Module(s) from study courses of the home institution** | |            |          |      |      | 10431 |

* From the Catalogue of Compulsory Electives a minimum of 3 modules must be completed with an examination (MOD2-05, MOD3-01 and MOD3-02). More than 18 credit points may be obtained which will be marked in the certificate.

** If compulsory elective modules of the Ruhr Master School (RMS) are part of the course programmes of Dortmund University of Applied Sciences and Arts (Fachhochschule Dortmund), students must complete the examinations within their own course programme.

Upon application, modules of the course programmes participating in the RMS may be elected.

*** At least 1 of the following Modules must be taken as an Elective: MOD-E04, MOD-E06, or MOD-E07.
Mathematics for Controls & Signals (MOD1-01) ......................................................... 7
Distributed and Parallel Systems (MOD1-02) ................................................................. 9
Embedded Software Engineering (MOD1-03) ................................................................. 11
Requirements Engineering (MOD1-04) ............................................................................. 14
Scientific & Transversal Skills Design (MOD1-05) .......................................................... 18
Mechatronic Systems Engineering (MOD2-01) ............................................................... 18
Microelectronics & HW/SW-Co-Design (MOD2-02) ....................................................... 20
R&D Project Management (MOD2-03) .............................................................................. 22
Signals & Control Systems 1 (MOD2-04) ........................................................................ 25
Research Project (Thesis) (MOD3-03) ............................................................................. 27
Master Thesis + Colloquium (P) ..................................................................................... 29
## Mathematics for Signals & Controls (MOD1-01)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10110/11</td>
<td>180 h</td>
<td>6</td>
<td>Sem. 1</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mathematics for Signals &amp; Controls</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

### 2 Course Description

This course introduces the necessary mathematical concepts for signal processing and control engineering. It starts with a tailored review of real and complex analysis. A major focus is on different kinds of integral transforms that are of essential use in subsequent courses. A huge amount of physical phenomena can be described by sets of linear differential equations and thus the latter are dealt with in this course. Linear algebra plays a prominent role in case of systems with several states and/or multiple inputs and outputs. Usually, sensor signals are corrupted by noise or other sources of uncertainty. To be able to deal with those, probability theory is introduced. Matlab and Octave are used as examples for state of the art tools for numerical mathematics and as a preparation for following courses.

### 3 Course Structure

1. Real and complex analysis
2. Fourier, Laplace and Z transform
3. Differential equations
4. Linear algebra
5. Probability theory
6. Introduction into Matlab/Octave
7. Numerical mathematics

### 4 Parameters

- Course characteristics: compulsory
- Course frequency: every year - winter semester
- Capacity: 25 students
- Course admittance prerequisites: none
- Skills trained in this course: theoretical, practical and methodological skills
- Assessment of the course: Written Exam (90 min) at the end of the course (100%)
- Teaching staff: Prof. Dr. Andreas Becker, (Prof. Dr. Thomas Felderhoff)

### 5 Learning outcomes

#### 5.1 Knowledge

- Knows basic theorems of complex analysis and linear algebra
- Knows relevant theoretical foundations of signal processing and control engineering
- Knows the most important concepts of probability theory

#### 5.2 Skills

- Can make use of analysis and linear algebra to describe physical phenomena
<table>
<thead>
<tr>
<th>5.3 Competence – attitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Can discuss mathematical prerequisites of mechatronic systems with experts</td>
</tr>
<tr>
<td>- Understands experts for mathematics and translates between different domains</td>
</tr>
</tbody>
</table>

### 6 Teaching and training methods
- Lectures & Exercises
- Labs with Matlab/Octave
- E-learning modules on higher mathematics, tool tutorials

### 7 Course mapping
Input for:
- MOD2-04 – Signals & Control Systems 1
- MOD-E04 – Signals and Systems for Automated Driving
- MOD-E06 – Computer Vision
- MOD-E07 – Signals & Control Systems 2

### 8 References
# Distributed and Parallel Systems (MOD1-02)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10120/21</td>
<td>180 h</td>
<td>6</td>
<td>Sem. 1</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed and Parallel Systems</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

## Course Description

Distributed systems are groups of networked computers and/or embedded systems, which have a common goal for their work. The terms distributed computing and parallel computing have a lot of overlap and frequently the term concurrent computing is used in this field. There is no clear distinction between them. This course is a prerequisite for the deeper understanding of multicore and manycore systems. It builds the theoretical core knowledge about cyber physical systems (CPS) and about the current state of research in the field of embedded distributed systems.

## Course Structure

1. Architectures for distributes systems (in principle)
2. Communication
   a. Synchronous, Asynchronous
   b. Peer-to-Peer, Broadcast, Multicast
   c. Protocols
3. Time and States
   a. States and Timestamps
   b. Clocks
4. Coordination and Agreement
   a. Transactions and Concurrency Control
   b. Deadlocks
   c. Replication and Fault Tolerance
5. Scheduling/Partitioning/Distribution (Multicore/Manycore)
6. Cyber physical systems (CPS)
7. Dependable Systems
8. Programming Paradigms and Methods

## Parameters

- Course characteristics: compulsory
- Course frequency: every year - winter semester
- Capacity: 25 students
- Course admittance prerequisites: computer science & programming
- Skills trained in this course: theoretical and methodological skills
- Assessment of the course: Written Exam (60 min) at the end of the course (50%) and individual homework (50%): paper/report about a recent topic from CPS research
- Teaching staff: Prof. Dr. Burkhard Igel, (Prof. Dr. Erik Kamsties)

## Learning outcomes

5.1 Knowledge

- Knows theory of distributed and parallel systems
- Knows critical issues concerning reliable distributed systems
- Knows recent research about partitioning and scheduling for cyber physical systems
5.2 Skills
- Can assess the feasibility of distributed CPS
- Can implement algorithms for distributed embedded systems
- Can model the behavior of distributed CPS
- Can apply state of the art tools and can develop new tools for distribution

5.3 Competence - attitude
- Can setup tooling and design flows
- Can discuss distribution issues with computer scientists
- Understands the potential of concurrency in CPS

6 Teaching and training methods
- Lectures & Exercises, AMALTHEA and TA tool labs
- e-learning modules on theoretical informatics, tool tutorials
- Presentation and discussion of an industry case by a partner company (e.g. Bosch, BHTC, TA)

7 Course mapping
Input for:
- MOD2-01- Mechatronic Systems Engineering
- MOD2-02 – Microelectronics & HW/SW Codesign
- MOD-E03 – SW Architectures for Embedded and Mechatronic Systems

8 References


Research Papers: Lamport, Chandy & Lamport

Other recent research papers
Embedded Software Engineering (MOD1-03)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10130/31</td>
<td>180 h</td>
<td>6</td>
<td>Sem. 1</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**
   - Embedded Software Engineering

2. **Contact hours**
   - 4 SWS / 60 h

3. **Self-Study**
   - 120 h

4. **Planned Group Size**
   - 25 students

2 **Course Description**

Embedded software engineering is a multidisciplinary approach for developing Solutions to complex engineering problems. The continuing increase in system complexity is demanding integrated engineering practices combining software engineering, control engineering, mechanical engineering, and electrical engineering. Therefore, modeling embedded systems often results in a mix of models from a multitude of disciplines. An integrated modelling approach is provided by SysML as an extension of the Unified Modeling Language (UML), version 2, which has become the de facto standard software modeling language. SysML is a robust language that addresses many of the embedded software engineering needs, while enabling the embedded software engineering community to leverage the broad base of experience and tool vendors that support UML. Embedded systems are often safety-critical applications where correct operation is vital to ensure the safety of the public and environment. Furthermore, these systems have to fulfill real-time requirements and they have to cope with restricted resources. Finally, we focus on several development processes of embedded systems and their underlying tools.

In addition to the lecture exercises are organized to give an insight how to use state of the art approaches and tools. Within small projects the students can contribute the gained knowledge by using these introduced tools and concepts.

3 **Course Structure**

1. Characteristics of Embedded (and real-time) Systems
2. Motivation for Embedded Software Engineering
3. Modeling of Embedded Systems
4. Overview and Architecture of SysML
   a. SysML: Requirements and Use Cases
   b. SysML: Basic Concepts
   c. SysML: Modeling Structure with Blocks
   d. SysML: Modeling Constraints with Parametrics
   e. SysML: Modeling Control Flow-Based Behavior with Activities
   f. SysML: Modeling Message-Based Behavior with Interactions
   g. SysML: Modeling Event-Based Behavior with State Machines
   h. SysML Tools in General and Enterprise Architect
5. Development Processes of Embedded Software Systems
6. SW Quality Management, Software-Test
7. Development Tools (e.g. Enterprise Architect, IBM Rational Rhapsody)

4 **Parameters**

- Course characteristics: compulsory
- Course frequency: every year - winter semester
- Capacity: 25 students
- Course admittance prerequisites: computer science & programming
Skills trained in this course: theoretical, practical and methodological skills
Assessment of the course: Written Exam (60 min) at the end of the course (50%) and group work as homework (50%) with Enterprise Architect or IBM Rhapsody use case and demonstration/presentation
Teaching staff: Prof. Dr. Stefan Henkler, (Prof. Dr. Martin Hirsch)

5 Learning outcomes

5.1 Knowledge
- Students know the characteristics of embedded (and real-time) systems
- Students know the most important SysML diagrams.
- Students know the syntax and semantic of the most important SysML diagrams.
- Students know modeling tools for embedded software systems.
- Students know processes and methods of embedded software engineering.

5.2 Skills
- Students can choose SysML-Diagrams to model specific software aspects.
- Students can model structural aspects by means of block diagrams.
- Students can model constraints by means of parametric diagrams.
- Students can model control flow-based behavior by means of activity diagrams.
- Students can model message-based behavior by means of interaction diagrams.
- Students can model event-based behavior by means of state machines.
- Student can tailor processes and methods to specific project needs.
- Students can evaluate and use tools for embedded Software engineering.

5.3 Competence - attitude
- Students develop an attitude to embedded software engineering according to modeling and processes.
- Students show a quality attitude according to embedded software engineering modeling.
- Students understand the main challenges of complex embedded software projects.
- Students understand the importance of modeling complex embedded software systems
- Students can improve their effectiveness and efficiency by using dedicated methods and tools to support engineering processes.
- Students understand the differences between software and embedded software systems projects and act accordingly.

6 Teaching and training methods
- Lectures introducing concepts, methods and tools
- Group work to train concepts and methods, to develop skills and to work on case studies
- Home work to add contributions on a case study as group work
- Presentations to communicate results
## Course mapping

Input for:
- MOD2-01 - Mechatronic Systems Engineering
- MOD2-02 – Microelectronics & HW/SW Codesign
- MOD-E03 – SW Architectures for Embedded and Mechatronic Systems
- MOD-E10 – Automotive Systems

Connects to:
- MOD1-02 - Distributed and Parallel Systems

## References


### Requirements Engineering (MOD1-04)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10140/41</td>
<td>180 h</td>
<td>6</td>
<td>Sem. 1</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Requirements Engineering</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

#### Course Description

Requirements engineering (RE) is the very first activity in software, systems, and service development. This course builds on software engineering skills from 1st semester (UML, SysML). Deriving a comprehensive set of requirements is a mandatory and critical task in the early phase of the systems engineering design flow. Requirements are the starting point and main angle for design, verification & validation, and for the test and integration of systems. Configuration and change request management are connected with RE. Defining requirements and dealing with requirements in a structured way is still a major area for research on tools and methodologies – especially for large and complex mechatronic systems. In this module, students will get specific knowledge about the state of the art and the main future challenges in Requirements engineering.

#### Course Structure

1. Introduction (What is a requirement?, problem vs. solution)
2. Frameworks (e.g. Jackson's WRSPM Model)
3. Requirements Engineering Process (stakeholder, activities)
4. System and system context
5. Elicitation of requirements (techniques and supporting activities, Kano model)
6. Textual requirements documents
7. Requirements modeling (e.g. goal-oriented modeling, requirements patterns)
8. Non-functional requirements
9. Validation of requirements
10. Requirements Management (attributes, prioritization, traceability, change management, RE tools, CMMI, ReqIF exchange format)
11. Software product lines and variability management

#### Parameters

- **Course characteristics:** compulsory
- **Course frequency:** every year - summer semester
- **Capacity:** 25 students
- **Course admittance prerequisites:** none
- **Skills trained in this course:** practical, methodological, and personal skills
- **Assessment of the course:** Paper/essay on literature review about recent research as individual homework (50%) and group work as homework (50%): DOORS demonstration and presentation of example
- **Teaching staff:** Prof. Dr. Erik Kamsties, (n.n.)
## Learning outcomes

### 5.1 Knowledge
- Knows frameworks and models for RE
- Knows relevant RE processes and interfaces to other processes
- Knows concepts and recent research on product line and variability management

### 5.2 Skills
- Can model requirements with RE tools
- Can set up and integrate RE tools into tool chains and design flows
- Can derive requirements in a structured and comprehensive way

### 5.3 Competence - attitude
- Understands the importance of RE in the early project phase
- Can set up and lead RE in a cross domain team

## Teaching and training methods

- Lectures introducing concepts, methods and tools
- Group work to train concepts and methods, to develop skills and to work on case studies
- Literature review and Essay writing
- Home work to add contributions on a case study as group work
- Presentations to communicate and demonstrate homework

## Course mapping

Module shared with Master Digital Transformation and Master Informatik

Input for:
- MOD-E10 – Automotive Systems

Connects to:
- MOD2-01 – Mechatronic Systems Engineering
- MOD2-03 – R&D Project Management

## References


## Scientific & Transversal Skills (MOD1-05)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10160/61</td>
<td>180 h</td>
<td>6</td>
<td>Sem. 1</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scientific &amp; Transversal Skills</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

### 2 Course Description

This module is tailored for new students with different levels of proficiency from their bachelor programmes. It is intended to close the gaps to the knowledge required for the master programme. Students select a minimum of 4 out of 7 compact courses on basic topics relevant for the further study programme. These compact courses will enable students with different backgrounds to get a smooth start into the master programme.

### 3 Course Structure

The programme offers a selection of about 7 compact courses. More compact courses might be added according to the needs of the individual student group:

1. Compact Programming Course (Java)
3. Embedded Systems Lab Project
4. Mini Project
5. Research Methods and Tools A (RMT-A)
6. Engineering Communication 1 (German)
7. Engineering Communication 1 (other language)

### 4 Case Studies

None – courses contain small labs

### 5 Parameters

- Course characteristics: compulsory, students have to choose a minimum of 4 out of 7 courses, based on assessment of their prior knowledge
- Course frequency: every year - winter semester
- Capacity: 25 students
- Course admittance prerequisites: none
- Skills trained in this course: methodological, practical and scientific skills
- Assessment of the course: tests (60 min) for each compact course, graded project work, compact course results are summarized for overall module grade
- Teaching staff: Prof. Dr. Rolf Schuster, professors + tutors for each compact course

### 5 Learning outcomes

#### 5.1 Knowledge
- Knows the foundations of each topic at least up to a bachelor level

#### 5.2 Skills
- Can apply the knowledge in the upcoming master courses

#### 5.3 Competence - attitude
- Can assess the gaps in own knowledge
- Can use a variety of tools, online-courses, tutorials to close the gaps through self-study
### 6 Teaching and training methods

- Lectures introducing concepts, methods and tools
- Labs to train practical skills
- Group work to train concepts and methods, to develop skills and to work on projects
- Literature review and essay writing
- Homework to contribute to projects as group work
- Presentations to communicate and demonstrate homework / project work

### 7 Course mapping

Input for: All other courses

### 8 References

- Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel: UML @ Classroom: An Introduction to Object-Oriented Modeling (Undergraduate Topics in Computer Science), Springer (2015)
Mechatronic Systems Engineering (MOD2-01)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10210/11</td>
<td>180 h</td>
<td>6</td>
<td>Sem. 2</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**
   - Mechatronic Systems Engineering

2. **Contact hours**
   - 4 SWS / 60 h

3. **Self-Study**
   - 120 h

4. **Planned Group Size**
   - 25 students

2 **Course Description**

Mechatronics Systems Engineering is both a challenge and a chance. A holistic and well elaborated engineering process for complex mechatronic system/cyber physical systems is a mandatory requirement for developing future intelligent products. Teaching this new school of engineering is the major goal of the whole master programme and an attractive offer for a university of applied sciences. This module introduces the holistic engineering methodology and offers the big picture for the other modules. The focus is on the early phase of mechatronic systems design since this phase offers the biggest leverage for better technical systems. Topics like cross domain engineering and systems integration are addressed, too. The content of the course is largely inspired from finding of the BMBF Spitzencluster “it’s OWL” and the new Fraunhofer Institute “Entwurfstechnik Mechatronik”. A continuous transfer of new findings into this course is intended.

3 **Course Structure**

1. Motivation:
   a. Examples for Mechatronic Systems
   b. Characteristics of Mechatronic Systems
   c. Challenges
2. Discpline-spanning development process
3. Systems Engineering (according to INCOSE SE handbook)
4. Conceptual Design of Mechatronic Systems
   a. CONSENS
5. The Software Engineering Domain
   a. MechatronicUML
   b. Behavior synthesis
7. Application to Use Case (Printing Industry, Rail Cab)

5 **Parameters**

- Course characteristics: compulsory
- Course frequency: every year - summer semester
- Maximal capacity: 25 students
- Course admittance prerequisites: mechanics/physics, basics of embedded systems
- Skills trained in this course: theoretical, practical and methodological skills
- Assessment of the course: Written Exam (90 min) at the end of the course (50%) and individual homework (50%): MechatronicUML model of an example
- Teaching staff: Prof. Dr. Stefan Henkler, (Prof. Dr. Martin Hirsch)

5 **Learning outcomes**

5.1 Knowledge

- Knows CONSENS, INCOSE SE handbook, MechatronicUML
- Knows mechatronic systems engineering processes
- Knows Enterprise Architect and other relevant tools
### 5.2 Skills
- Can model mechatronic systems
- Can apply methodology and state of the art tools on real use cases (e.g. printing machine)
- Can select tools and define tool chains and design flows

### 5.3 Competence - attitude
- Can structure the early phase of mechatronic systems design
- Can lead cross domain design of mechatronic systems
- Understands issues from different domains and can integrate solutions into a holistic design

### 6 Teaching and training methods
- Lectures, Labs (with Enterprise Architect and other tools), homework
- Access to tools and tool tutorials
- Access to recent research papers

### 7 Course mapping
**Input for:**
- MOD-E03 – SW Architectures for Embedded and Mechatronic Systems
- MOD-E08 – Formal Methods

**Requires:**
- MOD1-03 - Embedded Software Engineering

**Connects to:**
- MOD1-04 – Requirements Engineering
- MOD2-03 – R&D Project Management

### 8 References


INCOSE: Guide to the Systems Engineering Body of Knowledge - G2SEBoK:
http://g2sebok.incose.org/app/mss/menu/index.cfm
# Microelectronics & HW/SW-Co-Design (MOD2-02)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10220/21</td>
<td>180 h</td>
<td>6</td>
<td>Sem. 2</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Microelectronics &amp; HW/SW-Co-Design</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

## 2 Course Description

Digital Systems are the main hardware platform for embedded systems and the target of embedded SW development. A good knowledge and overview of available HW platforms is required. Furthermore, a concurrent engineering process (HW/SW CoDeign) is used to develop state of the art embedded systems. The coordination of (more agile) SW development and (more V-model) HW development is a challenge. Digital system development is applying complex tools and tool chains. The goal of this module is to enable to students to select, to assess, and to develop digital target platforms for embedded systems.

## 3 Course Structure

1. Microelectronic Components for Embedded Systems
   - a. DSP, Microcontroller
   - b. FPGA
   - c. ASIC, ASSP
   - d. Memories
   - e. Communication components (e.g. serial busses)
   - f. PCB and standard circuits
2. Digital systems design methodologies and processes
   - a. ESL concepts
   - b. SystemC
   - c. VHDL/Verilog
   - d. Simulation and validation
   - e. HW/SW partitioning
   - f. Verification and test
   - g. Synthesis (on FPGA)
3. Virtual Prototypes and HW/SW co-verification
4. Tools and Tool Chains
5. New Trends: Multicore/Manycore, SoC, 3D, MEMS

## 4 Parameters

- Course characteristics: compulsory
- Course frequency: every year - summer semester
- Capacity: 25 students
- Course admittance prerequisites: electronics, basics of embedded systems
- Skills trained in this course: theoretical, practical and methodological skills
- Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and group work as homework (50%): SystemC or VHDL implementation, mapping on FPGA, demonstration and presentation
- Teaching staff: Prof. Dr. Peter Schulz, (Prof. Dr. Carsten Wolff)

## 5 Learning outcomes

5.1 Knowledge

- Knows microelectronic components of embedded systems
- Knows digital systems design methodology and processes
- Knows tools and technologies for digital design
- Knows concept of virtual prototype and its application in HW/SW Codesign

### 5.2 Skills
- Can compose an embedded system out of microelectronic components
- Can describe digital systems with SystemC or VHDL
- Can run a digital simulation
- Can assess synthesis and verification reports for simple designs
- Can run test and debug sessions with FPGAs

### 5.3 Competence - attitude
- Can set up HW/SW Codesign projects for embedded systems
- Can choose and tailor the tool chain and methodology
- Can present and demonstrate the design flow for a digital design project

### 6 Teaching and training methods
- Lectures
- Labs with: SystemC and VHDL simulation (Mentor), FPGA synthesis (Mentor or Synopsis) and FPGA implementation (Xilinx or Lattice). Access to tools and tool tutorials (Europractice tool chain)

### 7 Course mapping

**Input for:**
- MOD-E09 – System on Chip Design

**Requires:**
- MOD1-03 - Embedded Software Engineering

**Connects to:**
- MOD2-03 - R&D Project Management

### 8 References

Documentation of Europractice – Mentor Graphics Tools and Cadence Tools


Clive “Max” Maxfield (Editor): “FPGAs World Class Designs”, Newnes / Elsevier, 2009


Schaumont, Patrick: A Practical Introduction to Hardware/Software Codesign. Springer 2010

Bailey, Brian, Martin, Grant: ESL Models and their Application: Electronic System Level Design and Verification in Practice. Springer 2010
R&D Project Management (MOD2-03)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10230/31</td>
<td>180 h</td>
<td>6</td>
<td>Sem. 2</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**
   - R&D Project Management

2. **Contact hours**
   - 4 SWS / 60 h

3. **Self-Study**
   - 120 h

4. **Planned Group Size**
   - 25 students

---

**Course Description**

The course R&D project management is focusing on processes, methods and tools for the management of innovative research and development projects in engineering. R&D projects are characterized by creativity and a high degree of innovation and uncertainty. Advanced project management methodology has to deal with the uncertainty and has to foster creativity. Apart from this general problem, R&D project methodology has to be aligned with the engineering processes and with the different engineering domains. Topics like quality management, configuration management and specific tools for risk management are part of the methodology, too. The course enables students to understand and structure R&D projects and to choose appropriate tools and methods based on a proper analysis of the project characteristics. The students are able to tailor the methodology and they understand the remaining gaps in the methodology. They can develop new project management methods and tools to fill the gaps and they can do research to assess the effectiveness and efficiency of project management methodology in R&D. The course is based on one main project case study and several small cases for specific topics.

---

3. **Course Structure**

1. Characteristics of R&D projects
2. Project Management Fundamentals
   a. Project Management Lifecycle / Project Lifecycle
   b. Roles and Structures
   c. Planning and Monitoring / Cost, Time, Scope, Quality
3. Integration Management
4. Scope Management
5. Schedule Management
6. Cost Management including Earned Value Analysis
7. Risk Management for R&D Projects
8. Project Resource Management
9. Stakeholder and Communication Management
10. Quality Management
11. Development Models and Special Approaches
    a. V-Model
    b. Agile Development
    c. Lean Management
12. Change and Configuration Management

---

4. **Parameters**

- Course characteristics: compulsory
- Course frequency: every year - summer semester
- Capacity: 25 students
- Course admittance prerequisites: none
- Skills trained in this course: methodological and personal skills
- Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and group work as homework (50%): project kickoff/release report and presentation
- Teaching staff: Prof. Dr. Carsten Wolff, (Dr. Oliver Hempel)
### Learning outcomes

5.1 Knowledge
- Students know the basic body of knowledge for project management
- Students know processes, methods and tools for risk management for R&D projects (e.g. risk register, risk mitigation)
- Students know processes, methods and tools for configuration management (esp. from SW engineering)
- Students know processes, methods and tools for change management
- Students know processes, methods and tools for quality management according to ISO9001 and TS16949
- Students understand the importance of Reviews in R&D projects
- Students understand the main challenges of large R&D projects

5.2 Skills
- Students can tailor processes and methods to the respective projects
- Students can apply the respective project management methodology
- Students can assess R&D projects and can extract relevant characteristics
- Students can develop new methods according to gaps in the existing methodology
- Students can do the complete planning and preparation of a real project case
- Students can develop relevant KPIs and scorecards for measuring effectiveness and efficiency

5.3 Competence - attitude
- Students develop an attitude to project management according to engineering standards
- Students show a quality attitude according to engineering standards
- Students manage projects based on structured and well defined processes and in depth analysis
- Students can achieve high effectiveness and efficiency in running complex and innovative R&D projects
- Students understand the differences between small and large projects and act accordingly

### Teaching and training methods

- Lectures introducing concepts, methods and tools
- Group work to train concepts and methods, to develop skills and to work on case studies
- Home work to add contributions on a case study as group work
- Presentations to communicate results
- Presentation and discussion of an industry case by a partner company

### Course mapping

Input for:
- MOD-E10 – Automotive Systems

Requires:
- MOD1-03 - Embedded Software Engineering

Connects to:
- MOD1-04 – Requirements Engineering
- MOD2-01 – Mechatronic Systems Engineering
- MOD2-02 – Microelectronics & HW/SW Codesign

### References


<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>DING, Ronggui</td>
<td>Key Project Management Based on Effective Project Thinking</td>
<td>Springer-Verlag</td>
<td>2016</td>
</tr>
<tr>
<td>Gerardi, Bart</td>
<td>No-Drama Project Management – Avoiding Predictable Problems</td>
<td>Apress</td>
<td>2011</td>
</tr>
</tbody>
</table>
### Course Title
Signals & Control Systems

### Contact hours
4 SWS / 60 h

### Self-Study
120 h

### Planned Group Size
25 students

### Course Description

Cyber-physical systems (CPS) interact with the physical world by making use of sensors and actuators. Their main source of information is a variety of signals. The analysis, processing and understanding of signals and the inherent information is a central topic for the development of mechatronic systems. This module delivers the theoretical foundations for the understanding of signal processing and control engineering problems and algorithms.

The description of sensors and actuators by linear time-invariant systems is a powerful tool for the description of the dynamic behavior of mechatronic systems. The corresponding concepts are dealt with in the first block of this course.

Control systems are the connection between the mechanical/physical world and the control task performed by the embedded system. The goal of this module is to enable students to interact with control system experts and to integrate their results into embedded and mechatronic systems by learning the basic principles of feedback and control engineering.

Embedded signal processing and control systems are based upon time-discrete calculations. Thus a major focus of this course is to deal with time discrete signals and systems and the transformation from continuous time to discrete time.

Filters play a major role in the processing of data and they are widely used in signal processing and control engineering tasks. The analysis starts with continuous time filters and then introduces canonical structures for FIR and IIR filters.

An additional goal is to teach the use of advanced tools for signal processing and control system design.

### Course Structure

1. Linear time-invariant systems
2. State variable models
3. Linear feedback and control systems
4. Sampling theorem and discrete Fourier-transform
5. Structures for discrete time systems
6. Continuous time and discrete time filters (FIR/IIR filter)
7. Applications of the above
8. Signal processing and control engineering with Matlab/Simulink

### Parameters

- Course characteristics: compulsory
- Course frequency: every year - winter semester
- Capacity: 25 students
- Course admittance prerequisites: higher mathematics
- Skills trained in this course: theoretical and methodological skills
- Assessment of the course: Written Exam (90 min) at the end of the course (100%)
- Teaching staff: Prof. Dr. Andreas Becker, (Prof. Dr. Jörg Thiem)

5 Learning outcomes

5.1 Knowledge
- Knows relevant theoretical foundations of signal processing and control theory
- Knows mathematical background of linear feedback controllers
- Is aware of critical limitations of discrete time signals and the impact of sampling
- Knows basic analogue and digital filters

5.2 Skills
- Can analyze systems and signals
- Can model linear feedback controllers for mechatronic systems
- Can apply and design digital filters

5.3 Competence - attitude
- Can discuss control system design for mechatronic systems with experts
- Can lead cross domain design of control systems
- Understands control system experts and translates between different domains

6 Teaching and training methods
- Lectures & Exercises, Matlab/Simulink labs
- e-learning modules on mathematics and control theory, tool tutorials

7 Course mapping

Input for:
- MOD-E04 - Signals and Systems for Automated Driving
- MOD-E06 – Computer Vision
- MOD-E07 – Signals & Control Systems 2

Requires:
- MOD1-01 - Mathematics for Signals & Controls

8 References

R. Bishop, R. Dorf: Modern Control Systems, Pearson Education, 2010
## Research Project (Thesis) (MOD3-03)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10330/31</td>
<td>540 h</td>
<td>18</td>
<td>Sem. 3</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Research Project (Thesis)</td>
<td>0 SWS / 40 h</td>
<td>500 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

### Course Description

The research project is intended to introduce students into scientific research work in a bigger context. Students will participate in one of the ongoing research projects. They will contribute with an own sub project. The starting point is the definition of the research questions they want to answer and the selection of the appropriate methodology. The students will plan and execute their project independently with regular review and consulting. They will summarize their finding in a research project thesis (project report). The research project will be a preparation for further work on the master thesis. The intention of the research project is to familiarize with the research methodology in a certain scientific field and to formulate the scientific state of the art and the research questions. The student proves the ability to execute own and independent research on master level and with a certain complexity.

### Course Structure

Students will select a topic from one of the ongoing projects in CPS and Embedded Systems. The will get individual consulting and feedback. During the semester the students will write a project thesis and present it in a colloquium at the end of the semester.

Excellent results are intended to be published and presented (oral or poster) at a conference (can be done in connection with the master thesis, too).

### Parameters

- Course characteristics: compulsory
- Course frequency: every year - winter semester
- Capacity: 25 students
- Course admittance prerequisites: none
- Skills trained in this course: theoretical, practical, methodological, and personal skills
- Assessment of the course: project thesis about own research in an ongoing project as individual homework + presentation in colloquium (30 min), (100%)
- Teaching staff: all professors

### Learning outcomes

#### Knowledge

- Knows state of the art in a certain scientific field
- Knows open research questions in this field
- Knows relevant literature
- Knows methodology and tools to execute project

#### Skills

- Can define and plan an own research project
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3 Competence - attitude</td>
<td>5.3 Competence - attitude</td>
</tr>
<tr>
<td>- Can apply appropriate research methodology</td>
<td>- Can apply appropriate research methodology</td>
</tr>
<tr>
<td>- Can create own research findings</td>
<td>- Can create own research findings</td>
</tr>
<tr>
<td>- Can describe project execution, methodology and findings in a scientific report</td>
<td>- Can describe project execution, methodology and findings in a scientific report</td>
</tr>
<tr>
<td></td>
<td>- Can run an own more complex scientific research project</td>
</tr>
<tr>
<td></td>
<td>- Masters uncertainty and unknown topics in new area</td>
</tr>
<tr>
<td></td>
<td>- Can present and defend results (in colloquium or at a conference)</td>
</tr>
</tbody>
</table>

6 **Teaching and training methods**

- Project Work
- Writing of a scientific report
- Presentations to communicate and discuss the findings
- E-learning course on scientific work and scientific writing
- Individual review and feedback on papers and presentations

7 **Course mapping**

Input for:
- P – Master Thesis + Colloquium

8 **References**

According to topic

Master Thesis + Colloquium (P)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>900 h</td>
<td>30</td>
<td>Sem. 4</td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**
   - Master Thesis + Colloquium

<table>
<thead>
<tr>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 SWS / 60 h</td>
<td>840 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

2. **Course Description**
   - The master thesis is intended for the students to show their ability for scientific research work in a bigger context. Students will participate in one of the ongoing research projects. They will contribute with their own sub project and with their own scientific results. The starting point is the definition of the research questions they want to answer and the selection of the appropriate methodology. The students will plan and execute their project independently with regular review and consulting. They will summarize their findings in a master thesis (scientific report). The intention of the master thesis is to apply the research methodology in a certain scientific field and to contribute their own findings to that scientific field. The student proves the ability to execute own and independent research on master level and with a certain complexity. Furthermore, the master thesis proves the ability to summarize and publish the results according to scientific standards.

3. **Course Structure**
   - Students will select a topic from one of the ongoing projects in CPS and Embedded Systems. They will get individual consulting and feedback. During the semester the students will write a master thesis and present it in a colloquium at the end of the semester.

   Excellent results are intended to be published and presented (oral or poster) at a conference.

4. **Parameters**
   - Course characteristics: compulsory
   - Course frequency: every year - summer semester
   - Maximal capacity: 25 students
   - Course admittance prerequisites: max. 1 module from semester 1 - 3 not finished.
   - Skills trained in this course: scientific, theoretical, practical, methodological, and personal skills
   - Assessment of the course: master thesis about own research in an ongoing project as individual homework + presentation in colloquium (30 min), (100%)
   - Teaching staff: all professors

5. **Learning outcomes**
   - **5.1 Knowledge**
     - Knows state of the art in a certain scientific field
     - Knows open research questions in this field
     - Knows relevant literature
     - Knows methodology and tools to execute project
     - Knows how to document new findings according to scientific standards
<table>
<thead>
<tr>
<th>5.2 Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can define and plan an own research project</td>
</tr>
<tr>
<td>• Can apply appropriate research methodology</td>
</tr>
<tr>
<td>• Can create own research findings</td>
</tr>
<tr>
<td>• Can describe state of the art, methodology and findings in a scientific report</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5.3 Competence - attitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can compare own findings with state of the art and do a critical discussion</td>
</tr>
<tr>
<td>• Can run an own scientific research project and create new findings</td>
</tr>
<tr>
<td>• Masters uncertainty and unknown topics in new area</td>
</tr>
<tr>
<td>• Can present and defend results (in colloquium or at a conference)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6 Teaching and training methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Project Work</td>
</tr>
<tr>
<td>• Writing of a scientific report</td>
</tr>
<tr>
<td>• Presentations to communicate and discuss the findings</td>
</tr>
<tr>
<td>• E-learning course on scientific work and scientific writing</td>
</tr>
<tr>
<td>• Individual review and feedback on papers and presentations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7 Course mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>None – can be based on research project thesis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8 References</th>
</tr>
</thead>
<tbody>
<tr>
<td>According to topic</td>
</tr>
</tbody>
</table>

ELECTIVES

Applied Embedded Systems (MOD-E01) ................................................................. 32
Smart Home & Smart Building & Smart City (MOD-E02) ................................. 33
SW Architectures for Embedded and Mechatronic Systems (MOD-E03) ......... 36
Signals and Systems for Automated Driving (MOD-E04) ............................ 38
IoT & Edge Computing (MOD-E05) ..................................................................... 39
Computer Vision (MOD-E06) ............................................................................... 42
Signals & Control Systems 2 (MOD-E07) ......................................................... 44
Formal Methods (MOD-E08) ................................................................................ 46
System on Chip Design (MOD-E09) ................................................................. 48
Automotive Systems (MOD-E10) ........................................................................ 50
Hardware Project (MOD-E11) .............................................................................. 51
Model Based Systems Engineering (MOD-E12) .................................................. 53
Software for Robots (MOD-E13) ........................................................................... 55
Embedded Systems Hardware Design and Rapid Prototyping (MOD-E14) ....... 57
Trends in Embedded and Mechatronic Systems (MOD-E15) ............................ 59
Research Seminar (S) .......................................................................................... 60
# Applied Embedded Systems (MOD-E01)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10401</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Applied Embedded Systems 1</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

## 2 Course Description

Applied embedded systems such as embedded controllers for industrial (i.e. robotics) applications are surrounded from sensors and actuators. Together with other embedded systems they can be groups of networked computers, which have a common goal for their work. This course gives an overview about the recent state of the art in embedded and cyber physical systems. Each semester, a selected CPS application will be analyzed in depth. This can be from robotic, energy, mobile communications or industrial scenarios (industry 4.0). The student will learn how to explore and structure a certain application domain and how to map the acquired skills and knowledge to that particular domain. CPS applications will be selected from recent research projects.

## 3 Course Structure

1. Introduction to the application domain
2. Characteristics of CPS in the application domain
3. Architectures for application specific CPS
   a. Standards
   b. Platforms and Frameworks
   c. Design methodology and processes
4. Domain specific languages (DSL) and applications
   a. DSL engineering
   b. Tools and Tool Chain Integration
5. Target Platforms and Code Generation
   a. Code generation
   b. Using real time operating systems (RTOS)

## 4 Parameters

- Course characteristics: elective
- Course frequency: every year - summer semester
- Capacity: 25 students
- Course admittance prerequisites: none
- Skills trained in this course: theoretical, practical and methodological skills
- Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and group work as homework (50%): modeling and target mapping of an example with AMALTHEA tools, demonstration and presentation
- Teaching staff: Prof. Dr. Burkhard Igel, (Prof. Dr. Carsten Wolff)

## 5 Learning outcomes

### 5.1 Knowledge

- Knows standards and platforms for specific domain
- Knows target systems
- Has acquired overview of target domain
5.2 Skills
- Can describe relevant characteristics and challenges of application domain
- Can model mechatronic systems for the domain
- Can apply methodology and state of the art tools on real use cases
- Can select tools and define tool chains and design flows

5.3 Competence - attitude
- Can structure a real mechatronic systems design project
- Can communicate and find solutions with domain experts
- Understands issues from application domains and can integrate solutions into a holistic design

6 Teaching and training methods
- Lectures, Labs (with AMALTHEA tools), homework
- Access to tools and tool tutorials
- Access to recent research papers

7 Course mapping
Requires:
- MOD1-02 – Distributed and Parallel Systems
- MOD1-03 - Embedded Software Engineering
Connects to:
- MOD-E03 – SW Architectures for Embedded and Mechatronic Systems
- MOD-E10 – Automotive Systems

8 References
Research papers of IDiAL institute and research group:
Specifically:
APP4MC: http://wiki.eclipse.org/APP4MC
KUKSA: https://www.eclipse.org/kuksa/
Smart Home & Smart Building & Smart City (MOD-E02)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10413</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. Course Title
   Smart Home & Smart Building & Smart City

2. Contact hours
   4 SWS / 60 h

3. Self-Study
   120 h

4. Planned Group Size
   25 students

2. Course Description

The digital transformation is a major driver for the change in people’s living environment. It affects the technical design of infrastructure systems, starting from people’s home via larger buildings and reaching up to systems like cities or districts. It covers home automation, energy and mobility systems and assistance systems. The course introduces the trends, developments and standards from the smart home, smart building and smart city domains and put them into the context of software and IoT systems. The aim is to enable students to develop larger software systems within the given context and to integrate them with other IoT and cloud systems. Therefore, it is intended to form a domain specific view on the digital transformation.

3. Course Structure

1. Smart home
   1.1 Home automation
   1.2 Standards and bus systems (e.g. KNX)
   1.3 OpenStack
   1.4 Ambient Assisted Living

2. Smart Building
   2.1 Building Information Systems (BIM)
   2.2 Safety and Security in Smart Buildings
   2.3 Facility Management and Smart Building

3. Smart City
   3.1 Smart City concepts and relevant trends
   3.2 Integration of Logistics, Energy, Supplies and Mobility
   3.3 Stakeholder and Citizen Involvement
   3.4 Case Study: Smart City Alliance Dortmund

4. Parameters

- Course characteristics: elective
- Course frequency: every year - summer semester
- Capacity: 25 students
- Course admittance prerequisites: none
- Skills trained in this course theoretical, practical and methodological skills
- Assessment of the course: Written Exam (60 min) at the end of the course (50%) and Individual programming task (50%): implementation of Smart System (or parts of it), demonstration of the results
- Teaching staff: Prof. Dr. Ingo Kunold, staff from IKT institute, guest lecturers from joint research projects
### 5 Learning outcomes

#### 5.1 Knowledge
- Knows relevant home automation systems and standards
- Know smart building concepts (e.g. BIM)
- Knows relevant trends and projects in Smart City
- Is aware of critical limitations, esp. safety and security issues

#### 5.2 Skills
- Can design concepts for smart home/smart building/smart city systems
- Can implement IoT, Cloud and SW components into such systems
- Can apply state of the art tools and systems (e.g. KNX)
- Can select IoT and cloud platforms according to smart home/building/city requirements

#### 5.3 Competence – attitude
- Can discuss smart home/building/city systems with experts
- Can lead cross domain design in this domain
- Can contribute within the Dortmund Smart City Alliance

### 6 Teaching and training methods
- Theoretical knowledge: e-learning modules on Smart Systems, tool tutorials
- Practical Skills: Projects, Labs & Exercises, small project with Smart Systems
- Scientific Competences: own research on Smart Systems

### 7 Course mapping
Module shared with Master Digital Transformation

Connects to:
- MOD-E01 – Applied Embedded Systems
- MOD-E05 – IoT & Edge Computing

### 8 References


SW Architectures for Embedded and Mechatronic Systems (MOD-E03)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10403</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. Course Title
   SW Architectures for Embedded and Mechatronic Systems

2. Course Description
   The ongoing complexity increase in mechatronic solutions consequently leads to more complex embedded systems and embedded software. Therefore, advanced SW engineering methodology from large software development projects is consecutively applied in the embedded world, too. Software architectures help to structure, to manage and to maintain large embedded SW systems. They allow re-use, design patterns and component based development. In addition, specific topics like safety, SW quality, integration and testing are addressed by SW architectures and respective standards (e.g. AUTOSAR). In this module, students learn about the concepts and structure of SW architectures for embedded systems.

3. Course Structure
   1. Characteristics of Embedded (and real-time) Systems
   2. Motivation for Architectures for Embedded and Mechatronic Systems
   3. Software Design Architecture for Embedded and Mechatronic Systems
   4. Patterns for Embedded and Mechatronic Systems
   5. Real-Time Building Blocks: Events and Triggers
   6. Dependable Systems
   7. Hardware's Interface to Embedded and Mechatronic Systems
   8. Layered Hierarchy for Embedded and Mechatronic Systems Development
   9. Software Performance Engineering for Embedded and Mechatronic Systems
   10. Optimizing Embedded and Mechatronic Systems for Memory and for Power
   11. Software Quality, Integration and Testing Techniques for Embedded and Mechatronic Systems
   13. Multicore Software Development for Embedded and Mechatronic Systems

4. Parameters
   - Course characteristics: elective
   - Course frequency: every year - summer semester
   - Capacity: 25 students
   - Course admittance prerequisites: programming, basics of embedded systems
   - Skills trained in this course: theoretical, practical and methodological skills
   - Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and individual homework (50%): paper/essay on a recent research topic, presentation
   - Teaching staff: Prof. Dr. Stefan Henkler, (Prof. Dr. Martin Hirsch)

5. Learning outcomes
   5.1 Knowledge
      - Knows concepts and structure of SW architectures for embedded systems
      - Knows standards and frameworks
      - Knows specific challenges (e.g. real time, functional safety)
5.2 Skills
- Can define requirements and features for a specific problem
- Can develop a SW architecture for a specific problem
- Can model SW architectures with state of the art tools
- Can apply SW architecture standards to structure a project

5.3 Competence - attitude
- Ensures quality and safety for embedded SW
- Can discuss and assess the advantages and disadvantages of different SW architectures
- Understands the main issues within research about SW architectures for embedded systems

6 Teaching and training methods
- Lectures, Labs (with AMALTHEA and Artop tools), homework
- Access to tools and tool tutorials
- Access to recent research papers
- Presentation of an industry case by partner BHTC GmbH

7 Course mapping
Requires:
- MOD1-02 – Distributed and Parallel Systems
- MOD1-03 - Embedded Software Engineering
- MOD2-01 – Mechatronic Systems Engineering

Connects to:
- MOD-E01 – Applied Embedded Systems
- MOD-E10 – Automotive Systems

8 References


Bruce P. Douglass, Real-Time Design Patterns: Robust Scalable Architecture For Real-Time Systems, Addison-Wesley, 2009

# Signals and Systems for Automated Driving (MOD-E04)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10404</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signals and Systems for Automated Driving</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

## Course Description

Automated driving requires the use of a multitude of sensors, controllers and actuators installed on the vehicle. Additionally, vehicle to vehicle and vehicle to infrastructure communication will be necessary. This course gives an overview about technologies used for automated driving. It starts with an overview about current R&D trends and then covers several sensor technologies with a special focus upon radar. Students will learn basic principles of stochastic signal processing and its application to tracking and mapping. Motion models and vehicle control technologies will be discussed to gain further insight into requirements for sensors and algorithms. Additional focus of this course is on architectures and infrastructures for automated driving. This includes bus interfaces and SW architectures as well as the basic principles of systems engineering. ISO 26262 as well as legal frameworks and their application to automated driving will be discussed. In addition to the lecture, exercises and small projects give additional insight into the technologies and concepts introduced in this course.

## Course Structure

1. Technology overview  
2. Sensors  
3. Signal processing  
4. Actuators & Vehicle Control  
5. System Architectures  
6. System Engineering  
7. ISO 26262  
8. Legal frameworks

## Parameters

- Course characteristics: elective  
- Course frequency: every year - summer semester  
- Capacity: 25 students  
- Course admittance prerequisites: higher mathematics, programming, signal processing  
- Skills trained in this course: theoretical, practical and methodological skills  
- Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and group work as homework (50%)  
- Teaching staff: Prof. Dr. Andreas Becker

## Learning outcomes

### 5.1 Knowledge

- Knows common driver assistance components and architectures  
- Knows basic signal processing algorithms for radars  
- Knows state estimation algorithms  
- Knows basics of related system engineering and norms

### 5.2 Skills

- Can develop tracking algorithms  
- Can develop radar signal processing algorithms  
- Can analyze requirements for subsystems of automated driving
### 5.3 Competence – attitude
- Understands the challenges in the development of automated driving and can discuss with experts from different domains
- Can lead development of subsystems for automated driving
- Can lead system level tests for automated driving

### 6 Teaching and training methods
- Lectures, Labs (with Matlab/Simulink)
- Access to tools and tool tutorials
- Access to recent research papers
- Company visit

### 6 Course mapping
Requires:
- MOD1-01 - Mathematics for Signals & Controls

Connects to:
- MOD1-04 – Requirements Engineering
- MOD2-01 – Mechatronic Systems Engineering
- MOD-E10 – Automotive Systems
- MOD-E06 – Computer Vision

### 7 References


Bar-Shalom et al., Estimation with Applications to Tracking and Navigation, John Wiley & Sons, 2001

Maurer et al., Automotive Systems Engineering, Springer 2013
# IoT & Edge Computing (MOD-E05)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10405</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th><strong>Course Title</strong></th>
<th><strong>Contact hours</strong></th>
<th><strong>Self-Study</strong></th>
<th><strong>Planned Group Size</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Internet of Things</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

## Course Description

Internet of things (IoT) is a fundamental building block for digitization and the upcoming information society. This course provides insights into key IoT-technologies including embedded systems, networks and cloud computing. For the selection of use cases and technologies the course focuses on the area of Edge Computing. Within this area student will learn about latency analysis and optimization in distributed systems. Last not least, the course offers hands on experiences with IoT and Edge Computing technologies through focused team projects and homework assignments.

## Course Structure

1. Introduction
2. Real-time Embedded Systems
3. Real-Time Networking
4. Cloud Computing
5. Edge Computing

## Parameters

- Course characteristics: elective
- Course frequency: every year - summer semester
- Capacity: 25 students
- Course admittance prerequisites: Basics in embedded systems, networks and programming
- Skills trained in this course: theoretical, practical and methodological skills
- Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and group work as homework (50%)
- Teaching staff: Prof. Dr. Rolf Schuster

## Learning outcomes

### 5.1 Knowledge
- Knows concepts and architectures of real-time embedded systems
- Knows key aspects of real-time networking
- Has acquired overview of cloud computing and selected cloud platforms

### 5.2 Skills
- Can implement, deploy and test simple IoT-systems
- Can set-up and utilize a cloud system
- Can analyze the E2E latency in distributed systems

### 5.3 Competence - attitude
- Can design a simple IoT system for a given set of requirements
- Can structure an IoT development project regarding function and time
- Can propose and implement measures to reduce latency in a distributed system
## 6 Teaching and training methods

- Lectures, group project, homework
- Access to tools and tool tutorials
- Access to recent research papers

## 7 Course mapping

Module shared with Master Digital Transformation

Requires:
- MOD1-02 – Distributed and Parallel Systems
- MOD1-03 – Embedded Software Engineering
- MOD1-05 – Scientific & Transversal Skills

## 8 References


Open Edge Computing Initiative: [https://www.openedgecomputing.org/](https://www.openedgecomputing.org/)
Computer Vision (MOD-E06)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10406</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**
   - Computer Vision

2. **Contact hours**
   - 4 SWS / 60 h

3. **Self-Study**
   - 120 h

4. **Planned Group Size**
   - 25 students

2. **Course Description**

   Computer Vision is both a basic technology and an application domain for mechatronic and embedded systems. It is used in automotive systems, robotics and biomedical systems. This module focuses on the use in mobile robots (e.g. autonomous driving, unmanned air vehicles) and industrial robots and biomedical applications (e.g. surgical robotics), since Dortmund University of Applied Sciences and Arts has established many research activities in these domains. Research topics from research centres (biomedical technology, pimes) and other key areas of the university are defining the content of this module. The module introduces the basic algorithms and components for computer vision and robotic vision systems. In addition, students will learn about the application of that knowledge in the specific domain. The course will involve topics from a recent research project.

3. **Course Structure**

   1. Introduction
   2. Position and Orientation
   3. Light and Color
   4. Image Creation
   5. Image Processing
   6. Feature Extraction
   7. Multiple Images
   8. Advanced Topics and Applications

4. **Parameters**

   - Course characteristics: elective
   - Course frequency: every year - summer semester
   - Capacity: 25 students
   - Course admittance prerequisites: higher mathematics, basics of embedded systems
   - Skills trained in this course: theoretical, practical and methodological skills
   - Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and group work as homework (50%): modeling and target mapping of an example with Matlab/Simulink, demonstration and presentation
   - Teaching staff: Prof. Dr. Jörg Thiem, (Dr. Roland Brockers)

5. **Learning outcomes**

   5.1 Knowledge
   - Knows standards and platforms for computer vision
   - Knows cameras, components, target systems
   - Has acquired overview of algorithms and methods

   5.2 Skills
   - Can model signal processing path for computer vision
   - Can apply methodology and state of the art tools for computer vision
   - Can adapt and modify/parameterize relevant algorithms
5.3 Competence - attitude
- Can structure a real computer vision project
- Can integrate cameras and vision modules into mechatronic systems
- Can analyze mechatronic systems and derive requirements for computer vision

6  Teaching and training methods
- Lectures, Labs (with MATLAB/Simulink), homework
- Access to tools and tool tutorials
- Access to recent research papers

7  Course mapping
Requires:
- MOD1-01 – Mathematics for Controls & Signals
- MOD1-03 - Embedded Software Engineering
- MOD2-04 – Signals & Control Systems 1

Connects to:
- MOD-E01 – Applied Embedded Systems
- MOD-E04 – Signals and Systems for Automated Driving
- MOD-E10 – Automotive Systems

8  References
## Signals & Control Systems 2 (MOD-E07)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10407</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Signals &amp; Control Systems 2</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>Course Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Control theory is one major part of the description of the dynamic behavior of mechatronic systems. Control systems are the connection between the mechanical/physical world and the control task performed by the embedded system.</td>
</tr>
<tr>
<td></td>
<td>This module extends the concepts from Signals &amp; Control Systems 1 (MOD2-04) to systems with states that are not directly measurable and/or noise corrupted. For this purpose, observer structures, estimation and adaptive signal processing concepts are reviewed. Emphasis is put on digital control and signal processing to path the way to embedded processing.</td>
</tr>
<tr>
<td></td>
<td>Based on those concepts, the linear quadratic controller is dealt with as one example to deal with noisy measurement and control signals. Furthermore, in order to incorporate control constraints, modern control strategies like model predictive control are studied.</td>
</tr>
<tr>
<td></td>
<td>The goal of this module is to enable students to interact with control system experts and to integrate their results into embedded and mechatronic systems under consideration of real-world constraints.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>Course Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. State Variable Feedback Control Systems</td>
</tr>
<tr>
<td></td>
<td>2. Optimal control</td>
</tr>
<tr>
<td></td>
<td>3. Robust Control Systems</td>
</tr>
<tr>
<td></td>
<td>4. Digital control</td>
</tr>
<tr>
<td></td>
<td>5. Adaptive Signal Processing</td>
</tr>
<tr>
<td></td>
<td>6. State estimation</td>
</tr>
<tr>
<td></td>
<td>7. Linear Quadratic Gaussian Control</td>
</tr>
<tr>
<td></td>
<td>8. Model Predictive Control</td>
</tr>
<tr>
<td></td>
<td>9. Applications of the above</td>
</tr>
<tr>
<td></td>
<td>10. Control Engineering with Matlab/Simulink</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Course characteristics: compulsory</td>
</tr>
<tr>
<td></td>
<td>- Course frequency: every year - winter semester</td>
</tr>
<tr>
<td></td>
<td>- Capacity: 25 students</td>
</tr>
<tr>
<td></td>
<td>- Course admittance prerequisites: higher mathematics</td>
</tr>
<tr>
<td></td>
<td>- Skills trained in this course: theoretical, practical and methodological skills</td>
</tr>
<tr>
<td></td>
<td>- Assessment of the course: Written Exam (60 min) at the end of the course (50%) and group work as homework (50%) with Matlab/Simulink use case and demonstration/presentation</td>
</tr>
<tr>
<td></td>
<td>- Teaching staff: Prof. Dr. Andreas Becker, (Prof. Dr. Jörg Thiem)</td>
</tr>
</tbody>
</table>
### 5 Learning outcomes

#### 5.1 Knowledge
- Knows relevant theoretical foundations of state variable compensators
- Knows concepts of optimal and robust control
- Knows approaches of adaptive signal processing and state estimation
- Knows concepts of predictive control

#### 5.2 Skills
- Can model complex control systems for mechatronic systems
- Can estimate states that are not measurable
- Can apply modern concepts like model predictive control
- Can select embedded system platforms according to controller requirements

#### 5.3 Competence - attitude
- Can discuss control system design and signal processing for mechatronic systems with experts
- Understands control system experts and translates between different domains
- Can lead cross domain design of control systems

### 6 Teaching and training methods
- Lectures & Exercises
- Matlab/Simulink labs
- Tool tutorials

### 7 Course mapping

Requires:
- MOD2-04 – Signals & Control Systems 1

Connects to:
- MOD-E04 – Signals and Systems for Automated Driving
- MOD-E06– Computer Vision

### 8 References

Stergiopoulos, Advanced Signal Processing, CRC Press, 2009
Kouvaritakis, Cannon, Model Predictive Control, Springer, 2015
R. Bishop, R. Dorf: Modern Control Systems, Pearson Education, 2010
Formal Methods (MOD-E08)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10408</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**
   - Formal Methods in Mechatronics

2. **Course Description**
   Software has become the driving force in the development of self-optimizing mechatronic systems. Such systems include hard-realtime coordination, which is realized by software, at the network level between distributed components as well as controllers which are more and more implemented by software. The communication goes beyond the use of system and environmental data from controllers. If necessary, complex status information about appropriate protocols and communication channels are exchanged, which themselves can massively influence the underlying behavior of the individual components. This development leads to extremely complex hybrid (discrete / continuous) software. In addition, self-optimizing mechatronic systems are often used in safety-critical environments. This enforces the use of formal verification techniques to ensure the correctness of specified properties.

   In the course concepts and methods for the modelling and verification of these mechatronic systems are introduced and formally described. In order to enable an efficient verification for such mechatronic systems, techniques like abstraction, decomposition as well as rule-based modelling are introduced. Here, these non orthogonal techniques are skillfully combined. One aim is to handle all models specified by all different domains. The presented approach for the model-based verification of mechatronic systems is massively characterized by the integration of efficient verification techniques for the different domains, based on their domain specific model-based knowledge.

3. **Course Structure**
   1. Motivation:
      a. What are Formal Methods?
      b. Why should we use Formal Methods?
      c. When in the overall development process should we use Formal Methods?
   2. Model Checking
   3. Testing
   5. Recent Research: literature review

4. **Parameters**
   - Course characteristics: elective
   - Course frequency: every year - winter semester
   - Capacity: 25 students
   - Course admittance prerequisites: programming
   - Skills trained in this course: theoretical and methodological skills
   - Assessment of the course: Written Exam (60 min) at the end of the course (50%) and group work as homework (50%): verification of an example, demo + presentation
   - Teaching staff: Prof. Dr. Martin Hirsch, (Prof. Dr. Stefan Henkler)

5. **Learning outcomes**
   5.1 Knowledge
   - Knows methodology of formal verification
   - Knows relevant theoretical background
- Knows specific requirements

### 5.2 Skills
- Can methods on use case
- Can model verification artefacts (e.g. properties)
- Can use MechatronicUML approach and tools

### 5.3 Competence - attitude
- Can research on state of the art and theoretical background
- Can demonstrate and discuss results in group
- Can structure scientific field and get overview

### 6 Teaching and training methods
- Lectures, Labs (with MechatronicUML), homework
- Access to recent research papers
- Literature review and discussion of results

### 7 Course mapping
**Requires:**
- MOD1-02 – Distributed and Parallel Systems
- MOD1-03 - Embedded Software Engineering
- MOD2-01 – Mechatronic Systems Engineering

**Connects to:**
- MOD-E03 – SW Architectures for Embedded and Mechatronic Systems

### 8 References
Spivey: The Z Reference Manual (http://spivey.oriel.ox.ac.uk/mike/zrm/zrm.pdf)
E. Clarke et al.: Model Checking, MIT Press


System on Chip Design (MOD-E09)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10409</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>Annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**
   System on Chip Design

2. **Contact hours**
   4 SWS / 60 h

3. **Self-Study**
   120 h

4. **Planned Group Size**
   25 students

2. **Course Description**
   This course introduces Systems on Chip with a strong focus on Multi- and Many-core Systems on Chip (SoC) The course deals both with the technology and the building blocks of SoCs and with the design process and tool chain. Complex SoCs are the basic hardware platform for embedded systems. Their development is a major area for research about tools, methodologies and development processes. ASIC development projects and tool chains are complex in size, technology and project structure. Students learn about the architecture and capabilities of SoCs and about the design flow.

3. **Course Structure**
   1. Main building blocks of SoCs
      a. IP-cores (processors, communication, memories, sources for IP-cores)
      b. on-chip communication (topologies, wishbone)
      c. system definition
      d. ESL: electronic specification language
      e. on-chip vs. off-chip memory
      f. debugging methodologies
   2. Multicore and Manycore architectures
      a. ASIP and Networks on Chip (NoC)
   3. ASIC design flow
      a. Design entry (VHDL)
      b. Pre-silicon verification
      c. Synthesis & technology libraries
      d. Layout and signal integrity
      e. Timing closure
      f. Power routing, clocks and resets
      g. Semiconductor test & production

4. **Parameters**
   - Course characteristics: elective
   - Course frequency: every year - winter semester
   - Capacity: 25 students
   - Course admittance prerequisites: programming, electronics
   - Skills trained in this course: practical and methodological skills
   - Assessment of the course: Written Exam (60 min) at the end of the course (50%) and group work as homework (50%): implementation of a CoreVA based design, demonstration and presentation
   - Teaching staff: Prof. Dr. Peter Schulz, (Prof. Dr. Carsten Wolff)

5. **Learning outcomes**
   5.1 Knowledge
   - Knows basic components of SoCs
   - Knows modern multicore/manycore architectures and ongoing research
   - Knows SoC design tools and tool chains
## 5.2 Skills
- Can develop an SoC from building blocks
- Can move a simple design through the whole tool chain
- Can select technology, constraints and layout

## 5.3 Competence - attitude
- Understands ASIC design flow
- Can consult on SoC selection and decision about SoC design
- Masters set up and configuration of complex ASIC design tool chains

### 6 Teaching and training methods
- Lectures, Labs (with Europractice tools), homework
- Access to tool chains and tool tutorials
- Access to recent research papers
- Visit at Bielefeld university (CITEC) and Intel Mobile Communications GmbH

### 7 Course mapping
Requires:
- MOD1-02 – Distributed and Parallel Systems
- MOD2-02 – Microelectronics & HW/SW-Codesign

Connects to:
- MOD-E11 – Hardware Project

### 8 References
Clive “Max” Maxfield (Editor): “FPGAs World Class Designs”, Newnes / Elsevier, 2009
### Automotive Systems (MOD-E10)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10410</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**  
   Automotive Systems

2. **Contact hours**  
   4 SWS / 60 h

3. **Self-Study**  
   120 h

4. **Planned Group Size**  
   25 students

### Course Description

Automotive systems are a major application domain for mechatronic and embedded systems. Due to the complexity and the specific requirements (e.g., safety) the domain specific engineering is well elaborated and leading edge in the embedded systems industry. The research centre pimes deals with various automotive partners and research projects. This course gives an overview about the recent state of the art in automotive systems and transfers recent findings into teaching. The student will learn how to explore and structure a certain automotive application and how to map the acquired skills and knowledge to that particular domain. Furthermore, the students will learn about domain specific standards, processes and frameworks.

### Course Structure

1. Automotive Standards: e.g. AUTOSAR, Quality Standards, Automotive Spice  
2. Automotive development processes  
3. Tools in Automotive Engineering (ML/SL, Doors, Enterprise Architect)  
4. Automotive Supply Chain  
5. Automotive Software Development  
6. Functional Safety  
7. Testing and Verification  
8. Product Qualification  
9. Application Examples  
10. AMALTHEA Methodology and Tool Chain

### Parameters

- Course characteristics: elective  
- Course frequency: every year - winter semester  
- Capacity: 25 students  
- Course admittance prerequisites: programming, basics of embedded systems  
- Skills trained in this course: theoretical, practical and methodological skills  
- Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and group work as homework (50%): set up of an automotive system development project, modeling and target mapping of an example with AMALTHEA tools, demonstration and presentation  
- Teaching staff: Prof. Dr. Carsten Wolff, (Prof. Dr. Erik Kamsties)

### Learning outcomes

5.1 **Knowledge**
- Knows standards and platforms for automotive systems  
- Knows target systems  
- Knows specific requirements (e.g., safety)  
- Has acquired overview of automotive application domain

5.2 **Skills**
• Can develop automotive software with the AMALTHEA tool chain
• Can model an automotive system according to standards
• Can select tools and define tool chains and design flows

5.3 Competence - attitude
• Can structure a real automotive system development project
• Can communicate and find solutions with automotive experts
• Ensures quality and safety of applications

7 Teaching and training methods
• Lectures, Labs (with AMALTHEA tools and Matlab/Simulink), homework
• Access to tools and tool tutorials
• Access to recent research papers
• Company visit at one of the partner companies (Bosch, BHTC)

8 Course mapping
Requires:
• MOD1-02 – Distributed and Parallel Systems
• MOD1-03 - Embedded Software Engineering
Connects to:
• MOD-E01 – Applied Embedded Systems
• MOD-E03 – SW Architectures for Embedded and Mechatronic Systems

9 References
Markus Maurer, Hermann Winner (Eds.): Automotive Systems Engineering, Springer, 2013
Research papers of IDiAL institute and research group:
Specifically:
APP4MC: http://wiki.eclipse.org/APP4MC
KUKSA: https://www.eclipse.org/kuksa/
## Hardware Project (MOD-E11)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10412</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Project</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

### Course Description

The aim of this course is to provide students with theoretical and practical experience in hardware engineering. Therefore, the students work in teams on real world tasks in cooperation with industry partners. The course focuses on the development of SoCs, FPGAs or Microcontroller based Embedded Systems. During the course, the students need to apply hardware engineering methodology and they need to use hardware engineering tool chains. In summary, the students implement the complete life cycle from requirements engineering to design over the development of a hardware system.

### Course Structure

The course is training hardware engineering skills by applying the following competences (from previous modules) within a realistic project (e.g. industry case):

- Circuit Design, especially for ASICs and PCBs
- Hardware Architecture Design
- Hardware Description Languages
- Hardware Testing and Component Verification
- Hardware Development Tool Chains (ASIC, FPGA or PCB)
  - Version control systems
  - Functional Modeling (e.g. VHDL, SystemC)
  - Verification and Simulation
  - Synthesis
  - Timing Analysis and Verification
  - Layout and Design Rule Check
  - Documentation
- Requirements Engineering
- Project management, project planning, quality management

### Parameters

- Course characteristics: compulsory
- Course frequency: every year - winter semester
- Capacity: 25 students
- Course admittance prerequisites: none
- Skills trained in this course: practical, methodological, and personal skills
- Assessment of the course: completion of the practical task with a demonstration + presentation in colloquium (30 min) (100%)
- Teaching staff: Prof. Dr. Peter Schulz, (Prof. Dr. Carsten Wolff)
<table>
<thead>
<tr>
<th>5</th>
<th>Learning outcomes</th>
</tr>
</thead>
</table>
| 5.1 Knowledge | • Students know development tools for hardware design  
• Students know concepts and processes for hardware development  
• Students know how to create test beds for hardware testing  
• Students know hardware description languages (HDL), e.g. VHDL |
| 5.2 Skills | • Students can apply processes and methods to specific project needs  
• Students can evaluate and use tools for developing hardware systems in a team  
• Students can use tools to support the development process in a team  
• Students can use tools to verify and test hardware |
| 5.3 Competence – attitude | • Can discuss and defend results in topics related to the lecture content  
• Can work in a team on scientific topics  
• Can understand lecture related content and translates between different domains |
| 6 | Teaching and training methods |
| | • Practical development projects (in the Chiplab)  
• Tutorials for tools and processes  
• Joint team reviews and team meetings  
• Presentations to communicate and discuss the findings  
• Individual review and feedback on the project |
| 7 | Course mapping |
| | Requires:  
• MOD1-02 – Distributed and Parallel Systems  
• MOD2-02 – Microelectronics & HW/SW-Codesign |
| | Connects to:  
• MOD-E09 – System on Chip Design |
| 8 | References |
| | Europractice tools documentation (online) |
# Model Based Systems Engineering (MOD-E12)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10415</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Planned Group Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Based Systems Engineering</td>
<td>4 SWS / 60 h</td>
<td>120 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

## 2 Course Description

The demands on automotive computing platforms are continuously rising due to the increasing amount of software that is driven by new automotive functionalities. Deploying these applications to computing platforms will introduce several challenges, such as maintaining freedom from interference in safety-critical applications -as required by the ISO~26262 standard,- or meeting constraints such as timing requirements. As the complexity of those systems results in intricate and unforeseen impacts of product and project decisions on the system level, even in late development phases, an early assessment of design decisions will be a key factor for success.

This course gives an overview about the recent state of the art in model based systems engineering with focus on the emerging trends in automotive systems and transfers recent findings into teaching. The student will learn how to explore and structure models of automotive systems – especially in the context of hardware/software co-design – and how to map the acquired skills and knowledge to that particular domain. Furthermore, the students will learn about developing and integrating own rudimentary tooling into the APP4MC platform.

## 3 Course Structure

1. Trends and challenges for future automotive E/E architectures
2. Automotive Standards: e.g. AUTOSAR, EastADL, Amalthea, …
3. Eclipse APP4MC
4. Modelling embedded systems
5. Developing rudimentary tooling for analyzing resp. modifying existing models of applications
6. Open Source and proprietary tools in Model based Automotive Engineering (e.g. Vector / TA Tool Suite, Inchron, Eclipse APP4MC Task Visualizer, …)
7. Deploying software to embedded multi- and many-core hardware
8. Code generation
9. Testing and Verification
10. Application Examples

## 4 Parameters

- ECTS: 6
- Hours of study in total: 180
- Weekly hours per semester: 4
- Contact hours: 60
- Self-Study hours: 120
- Course characteristics: elective
- Course frequency: every year - winter semester
- Maximal capacity: 25 students
- Course admittance prerequisites: programming skills (pref. Java), basics of embedded systems
5 Learning outcomes

5.1 Knowledge
- Knows typical challenges in developing future e.g. automotive embedded systems and how to address these using model based approaches
- Knows how to apply 3rd party tools in MBSE
- Has acquired an overview on the various views on automotive applications

5.2 Skills
- Can model a system from e.g. the automotive context (software, hardware, global functionality) according to a real-world example
- Can assess an application using tools based on their model description
- Can select and develop own (rudimentary) tools as well as integrate these into design flows

5.3 Competence - attitude
- Can structure a real model based systems engineering development project
- Can communicate and find solutions with domain experts
- Understands challenges in using heterogeneous hardware platforms

6 Teaching and training methods
- Lectures, Labs (with APP4MC and 3rd party tools), homework
- Access to tools and tool tutorials from industrial partners (e.g. Bosch, Inchron, Vector)
- Access to recent research papers
- Company visit from at least one of the partner companies

7 Course mapping
Requires:
- MOD1-02 – Distributed and Parallel Systems
- MOD1-03 - Embedded Software Engineering
Connects to:
- MOD-E01 – Applied Embedded Systems

8 References
APP4MC Documentation: https://www.eclipse.org/app4mc/documentation/
Research papers in the context of model based systems engineering in the context of automotive development: https://scholar.google.de/citations?hl=en&user=iBKd0uAAAAAJ
Software for Robots (MOD-E13)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10416</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. Course Title
   Software for Robots

2. Contact hours
   4 SWS / 60 h

3. Self-Study
   120 h

4. Planned Group Size
   21 students

2 Course Description

Robotic systems are usually very complex and utilize extensive functions as well as a high amount of actuators, sensors, and software-algorithms. The development and maintenance of software for such a robotic system is a challenge for developers and requires robotic specific domain knowledge. As the field of robotics ranges from enormous industry robots to small consumer robots, this course focuses on (small) low-cost mobile robots. Therefore a demonstration platform, the S4R rover is used to introduce students to typical challenges and applications for mobile robots. The course gives an overview of current trends and research fields for mobile robots and will focus on hand-on sessions to develop their software solutions. The student will learn to develop, implement, and test the software for the S4R rover in small student groups within the lecture and practice sessions. Individual homework assignments give students a more in-depth knowledge of relevant research topics..

3 Course Structure

1. Introduction to mobile robotics
2. Introduction to the App4MC/ S4R rover
   * Hardware
   * Rover API
   * ROS (Robot Operating System) integration
3. Implementation of Computer Vision tools/ methods/ algorithms
4. Implementation of Navigation and Mappings tools/ methods/ algorithms
5. Application/ Use-Case definition and Implementation in small groups
6. Test and Verification
7. Presentation of Applications/ Use-Cases
8. Homework definition
9. Homework presentation

4 Parameters

- Course characteristics: elective
- Course frequency: every year - summer semester
- Capacity: 21 students (3 students per demonstrator (7))
- Course admittance prerequisites: programming skills (C/C++)
- Skills trained in this course: theoretical, practical and methodological skills
- Assessment of the course: Oral Exam at the end of the course (50%) and group work as homework (50%): Implementation of the software for a given mobile robot, testing software on hardware, development and implementation of a demonstration application, demonstration and presentation
- Teaching staff: Uwe Jahn, Prof. Dr. Christof Röhrig

5 Learning outcomes

5.1 Knowledge
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2 Skills</td>
<td></td>
</tr>
</tbody>
</table>
| Knows typical challenges in developing software for mobile robots  
| Knows how to use sensor and actuators on mobile robots  
| Knows how to use computer vision, navigation and mapping tools/methods/algorithms  
| Can select and integrate typical tools used in robotics within software development projects  
| Can implement software for mobile robots  
| Can test and verify applications for mobile robots  
| 5.3 Competence - attitude |  
| Can structure robotic systems design project  
| Can communicate and find solutions with domain experts  
| Understands issues from the robots application domains and can integrate solutions into a holistic design  
| 6 Teaching and training methods |  
| Lectures, Practice, homework  
| Access to tools and tool tutorials  
| Access to mobile robots demonstrators (7)  
| Access to recent research papers  
| 7 Course mapping |  
| Requires:  
| MOD1-02 - Distributed and Parallel Systems  
| MOD1-03 - Embedded Software Engineering  
| Connects to:  
| MOD-E01 - Applied Embedded Systems  
| MOD-E03 - SW Architectures for Embedded and Mechatronic Systems  
| MOD-E06 - Computer Vision  
| 8 References |  
| Robotics, Vision and Control, Peter Corke (ISBN 978-3-319-54413-7)  
| Embedded Robotics, Thomas Bräunl (ISBN 978-3-540-70534-5)  
### Embedded Systems Hardware Design and Rapid Prototyping (MOD-E14)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10417</td>
<td>180 h</td>
<td>4</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

#### 1 Course Title
- Embedded Systems Hardware Design and Rapid Prototyping
- Contact hours: 4 SWS / 60 h
- Self-Study: 120 h
- Planned Group Size: 20 students

#### 2 Course Description
This course covers all the steps from an idea to a working embedded system prototype. Rapid prototyping of embedded systems and electronic circuits in general is an essential tool in research and product development, because designing and prototyping is a cycle that is usually iterated a few times and should therefore be as fast as possible. Furthermore, the insights which result from rapid prototyping can directly go into the next design cycle. This course applies a project-based learning approach, where every student designs his own embedded system from schematic to layout. The complexity of the project can vary according to prior knowledge and experience of the individual student – it can be for example a simple 4-layer 32-bit microcontroller design using an ARM cortex M3 or a very complex 6 layer design using a Xilinx Zynq device, which is an integrated System on Chip and FPGA. After the layout is done the printed circuit boards (PCBs) will be manufactured externally. The students will then perform assembly and testing of their prototype. The practical lab work will be accompanied by lectures that present the theoretical foundations, which are necessary to create a good design and solve problems quickly. The presented topics include, principles of signal and power integrity of high-speed embedded systems, compliance measurements of modern interfaces like gigabit ethernet and EMI precompliance testing.

#### 3 Course Structure
1. Introduction to schematic design tools
2. Schematic design of an embedded system *(homework + presentation)*
3. Introduction to layout design tools
4. Principles of signal and power integrity
   a. Target Impedance
   b. Decoupling capacitors
   c. Power planes
   d. Impedance and length matching of traces for high speed signals
5. Microstrip antennas
6. Layout of an embedded system *(homework + presentation)*
7. Soldering techniques (classical, hot air, reflow)
8. Prototype assembly *(lab work)*
9. Hardware debugging techniques using modern measuring equipment
10. Testing and validation of embedded systems *(lab work)*
   a. Code generation to activate peripherals for testing
   b. Compliance testing of peripherals (i.e. Ethernet, DDR3, Bluetooth)
11. Theoretical fundamentals of EMI precompliance testing
   a. Conducted emissions according to CISPR standards
   b. Radiated emissions according to CISPR standards
   c. Measurement methods (Antenna, LISN)
12. EMI precompliance testing (emissions) using a spectrum analyzer *(lab work)*

#### 4 Parameters
<table>
<thead>
<tr>
<th>Course characteristics: elective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course frequency: every year - summer semester</td>
</tr>
<tr>
<td>Capacity: 20 students</td>
</tr>
<tr>
<td>Course admittance prerequisites: none</td>
</tr>
<tr>
<td>Skills trained in this course: theoretical, practical and methodological skills</td>
</tr>
<tr>
<td>Assessment of the course: Oral presentation (10 min) at the end of the course (50%) and results from homework/lab work (50%)</td>
</tr>
<tr>
<td>Teaching staff: Prof. Dr. Benjamin Menküç</td>
</tr>
</tbody>
</table>

### 5 Learning outcomes

#### 5.1 Knowledge
- Knows principles of schematic and layout design for embedded systems
- Knows theoretical foundations of power- and signal integrity
- Knows theoretical foundations and norms required for EMI precompliance testing

#### 5.2 Skills
- Can create a schematic of an embedded system using modern design tools
- Can create a layout of an embedded system while applying signal and power integrity principles
- Can assemble a PCB prototype with SMD components using different soldering techniques
- Can perform hardware debugging using modern measuring equipment
- Can perform compliance testing of high-speed interfaces
- Can perform EMI precompliance testing

#### 5.3 Competence - attitude
- Can break down a complex task into work packages and meet deadlines
- Can communicate and find solutions with domain experts
- Can present project status and results to an audience

### 6 Teaching and training methods

- Lectures, lab work, homework
- Access to modern measuring equipment (oscilloscope, vector network analyzer)
- Access to recent research papers

### 7 Course mapping

Connects to:
- MOD1-03 - Embedded Software Engineering
- MOD1-02 – Distributed and Parallel Systems
- MOD-E03 – SW Architectures for Embedded and Mechatronic Systems
- MOD-E10 – Automotive Systems

### 8 References


High-Speed Circuit Board Signal Integrity, Thierauf, Artech house (2017)


KiCad Like a Pro, Dalmaris, Tech Explorations (2018)
Trends in Embedded and Mechatronic Systems (MOD-E15)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10418</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

1. **Course Title**
   - Trends in Embedded and Mechatronic Systems

2. **Contact hours**
   - 4 SWS / 60 h

3. **Self-Study**
   - 120 h

4. **Planned Group Size**
   - 25 students

---

2. **Course Description**
   The module will introduce and discuss recent topics from scientific research and industrial R&D. The goal is to make students familiar with the trends and to encourage own scientific and practical work in the respective field. The module will use presentations by scientists and practitioners to introduce topics. Literature work including structured literature reviews and discussion of relevant research papers will further enhance the practical knowledge. Industry presentations and visits can deliver practical insights. The module can introduce several different areas or topics, or it can dive deep into one topic. This can involve own research work of students, e.g. in order to develop a research paper for a conference (preferably the Dortmund International Research Conference). The module can also include practical labs or experiments. Individual project work or group work in small project teams can be used to develop new results. Presentations can be used to discuss the results.

3. **Course Structure**
   10. Introduction of a new trend in Embedded and Mechatronics Systems
   11. Literature research and discussion of the state of the art
   12. (optional) company visit and/or discussion of practical cases
   13. Industry presentations
   14. Tool trainings and practical labs
   15. Own research, e.g. with experiments or projects
   16. Presentation of the results
   17. Preparation of a paper for a conference

4. **Parameters**
   - Course characteristics: elective
   - Course frequency: every year – summer/winter semester
   - Capacity: 25 students
   - Course admittance prerequisites: none
   - Skills trained in this course: theoretical, practical and methodological skills
   - Assessment of the course: Oral Exam (30 min) at the end of the course (50%) and group work as homework (50%): research on a recent technology trend
   - Teaching staff: depends on topic, (organizer: Prof. Dr. Carsten Wolff)

5. **Learning outcomes**
   5.1 Knowledge
   - Knows recent trends in Embedded and Mechatronic Systems
   - Knows the relevant scientific literature
- Knows practical cases

### 5.2 Skills
- Can do a structured literature review on a given topic
- Can design own research on the topic
- Can present research results

### 5.3 Competence - attitude
- Can systematically explore a new scientific field
- Can organize research work in an unknown field
- Can synthesize and summarize findings in a meaningful way
- Shows curiosity in scientific research

### 6 Teaching and training methods
- Lecturers and industry presentations
- Individual literature research
- Assignments, e.g. writing of a paper

### 7 Course mapping
**Requires:**
- Scientific & Transversal Skills (MOD1-05)

**Connects to:**
- Research Seminar
- Research Project (Thesis) (MOD3-03)
- Master Thesis and Colloquium

### 8 References
- Specific for the recent research topic
# Research Seminar (S)

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10411</td>
<td>180 h</td>
<td>6</td>
<td></td>
<td>annually</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th><strong>Course Title</strong></th>
<th><strong>Contact hours</strong></th>
<th><strong>Self-Study</strong></th>
<th><strong>Planned Group Size</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Research Seminar</td>
<td>1 SWS / 30 h</td>
<td>150 h</td>
<td>25 students</td>
</tr>
</tbody>
</table>

## 2 Course Description

The research seminar is intended to introduce students into scientific writing, literature review and into discussion of research questions in a scientific auditory. Students will write a scientific report or essay on a recent research topic from one of the ongoing projects. The seminar will be a preparation for further work on the research project thesis and the master thesis. The intention of the seminar is to explore a certain scientific field and to formulate the scientific state of the art and the open research questions. A motivation for students will be the possibility to publish and present excellent papers at a small conference.

Instead of the seminar and the homework, the students can attend a third elective module.

## 3 Course Structure

Scientific Methodology is taught with a 3 days intensive course "Research Methods and Tools B (RMT-B)" which students attend together with students from other Master programmes.

Students will select a topic from one of the ongoing projects in CPS and Embedded Systems. The will get individual consulting and feedback. During the semester the students will write a paper/report and present it in a colloquium at the end of the semester.

Excellent papers will be published and presented (oral or poster) at the Dortmund International Research Conference at FH Dortmund.

## 4 Parameters

- Course characteristics: compulsory
- Course frequency: every year - summer semester
- Capacity: 25 students
- Course admittance prerequisites: none
- Skills trained in this course: scientific, theoretical, methodological, and personal skills
- Assessment of the course: exam (60 min) for RMT-B (40%), Paper/essay on literature review about recent research as individual homework + presentation (30 min) in colloquium (60%)
- Teaching staff: all professors

## 5 Learning outcomes

### 5.1 Knowledge
- Knows state of the art in a certain scientific field
- Knows open research questions in this field
- Knows relevant literature

### 5.2 Skills
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5.3 Competence - attitude</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Can analyze scientific literature based on a comprehensive review</td>
</tr>
<tr>
<td></td>
<td>Can write a paper/report according to scientific standards</td>
</tr>
<tr>
<td></td>
<td>Can synthesize findings in own words</td>
</tr>
<tr>
<td></td>
<td>Can run an own small scientific research project</td>
</tr>
<tr>
<td></td>
<td>Can present and defend results at a conference</td>
</tr>
</tbody>
</table>

6 Teaching and training methods
- Course (Lecture) on Research Methods and Tools B (RMT-B)
- Literature review and Essay writing
- Presentations to communicate and discuss the findings
- E-learning course on scientific work and scientific writing
- Individual review and feedback on papers and presentations

7 Course mapping
Input for:
- MOD3-03 – Research Project (Thesis)
- P – Master Thesis + Colloquium

8 References
German and European Research Agendas, recent research papers