Cyber-Physical System Fundamentals					AR-215
Rota annually SS	**Duration** 1 Semester	**Semester** 2nd (Semester)	**SWS** 4 SWS	**Credit Points** 6	**Workload** 180 h

1 Modul structure

Course (Abbreviation)	Type/ SWS	Presence	Self study	Credits
a) Cyber-Physical System Fundamentals (CPSF)	Lecture/ 4 SWS	60 h	60 h	4
b) Cyber-Physical System Fundamentals (CPSF)	Lab	60 h		2

2 Language: English

3 Content:

The course is based on the presenter's book on the subject and includes the following topics:

1. Introduction: Definition of terms, scope of the course
2. Specification and modeling: models of computation, communication models, finite state machines, data flow, discrete event models, von-Neumann-models, expressiveness of models
3. CPS hardware: hardware-in-the-loop, sampling and A/D-conversion, processing, field-programmable gate arrays (FPGAs), communication hardware, D/A-conversion, sampling theorem, output
4. Standard software: embedded operating systems, real-time operating systems, priority inversion, middleware
5. Evaluation and validation: objective functions, Pareto-optimality, worst-case execution time, energy consumption, reliability, real-time calculus, verification
6. Mapping of applications to execution platforms: standard optimization techniques, real-time scheduling, rate monotonic scheduling, earliest deadline first scheduling, hardware/software partitioning, mapping of applications to heterogeneous multiprocessors
7. Selected optimizations.

Literature:

- Peter Marwedel: Embedded System Design – Cyber Physical System Fundamentals, Springer, 2010
- Lego Mindstorm NXT Technical documentation
- Technical documentation fort he used finite state machine design tool (StateMate or similar)

4 Goals

Students successfully finishing the course should be able to

- Understand hoe cyber-physical (CPS) hardware interacts with CPS software and use this knowledge to design CPS software,
- Select models of computation and programming languages that are appropriate for a given design problem,
- Select an appropriate scheduling technique for embedded systems,

Apply hardware/ software codesign techniques in order to optimize the systems which they are supposed to design.

5 Examination Requirements

The students have to pass both the lab and the finals.

6 Formality of Examination

☒ Module Finals ☐ Accumulated Grade

7 Module Requirements (Prerequisites)

Basic knowledge in programming as well as finite-state machines.

8	**Allocation to Curriculum:** Program: Automation & Robotics, Field of study: Robotics, Process Automation, Cognitive Systems,
9	**Responsibility/ Lecturer** *Prof. Dr. Peter Marwedel/* Prof. Dr. Peter Marwedel