<table>
<tr><th colspan="5">Cyber-Physical System Fundamentals</th><th>AR-215</th></tr>
<tr><td>Rota
annually SS</td><td>Duration
1 Semester</td><td>Semester
2nd (Semester)</td><td>SWS
4 SWS</td><td>Credit Points
6</td><td>Workload
180 h</td></tr>
<tr><td>1</td><td colspan="5">Modul Structure
<table>
<tr><th>Course (Abbreviation)</th><th>Type/ SWS</th><th>Presence</th><th>Self Study</th><th>Credit Points</th></tr>
<tr><td>a) Cyber-Physical System Fundamentals (CPSF)</td><td>Lecture/ 4 SWS</td><td>45 h</td><td>75 h</td><td>4</td></tr>
<tr><td>b) Cyber-Physical System Fundamentals (CPSF)</td><td>Lab</td><td colspan="2">60 h</td><td>2</td></tr>
</table></td></tr>
<tr><td>2</td><td colspan="5">Language: English</td></tr>
<tr><td>3</td><td colspan="5">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)</td></tr>
<tr><td>4</td><td colspan="5">Competencies
Students successfully finishing the course should be able to
• Understand how 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 design techniques in order to optimize the systems which they are supposed to design.</td></tr>
<tr><td>5</td><td colspan="5">Examination Requirements
The students have to pass both the lab and the finals.</td></tr>
<tr><td>6</td><td colspan="5">Formality of Examination
☒ Module Finals ☐ Accumulated Grade</td></tr>
<tr><td>7</td><td colspan="5">Module Requirements (Prerequisites)
Basic knowledge in programming as well as finite-state machines.</td></tr>
</table>

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