Embedded Systems Engineering

1. Semester

Subject to Approval by the Relevant Bodies

Course	Type	THW	ECTS
Analog Circuit Design	IC	3	5
Digital Circuit Design	IC	3	5
Electronic Packaging	IC	2	3
Design Tools and Laboratory Engineering	IC	3	3
Software Design and Architectures	IC	3	5
Data Analysis	IC	3	5
System Requirements Engineering	IC	2	3
Intercultural Communication	SE	1	1
		20	**30**

IC Integrated course

SE Seminar

THW Hours per week

ECTS European Credit Transfer and Accumulation System

Analog Circuit Design 3 SWS/5 ECTS

Teaching Content

- Stochastic signal theory: RMS, mean value, autocorrelation, cross correlation, spectral power density
- Noise in electronic circuits: noise types, noise sources, transistor noise, noise figure
- Circuit analysis with noise estimation: transistor and OpAmp noise calculations
- Active filter circuits: filter characteristics, Sallen-Key and multifeedback circuit, 4th order bandpass, NIC, Gyrator, gm-C filter, double-T
- Switched capacitor filter: charge balancing, amplifier, integrator, filter
- ADC/DAC specifications: ADC noise, ENOB, gain and offset error, INL, DNL, SFDR
- A/D and D/A converter: resistor-string, R-2R ladder, weighted current sources, charge redistribution, PWM, Delta-Sigma DAC, Flash, pipelined, SAR, Dual-Slope, Delta-Sigma ADC
- Bandgap references: base-emitter voltage temperature drift, bandgap ref. circuits
- Frontend design: example design with amplifier, filter and ADC, SNR and ENOB calculation
- Computer simulations of circuit noise

Competence Acquisition

After finishing this course, students can

- analyse the effect of noise in analogue circuits,
- design active filters for a specific specification,
- analyse the effect of component tolerances on analogue circuits,
- analyse switched capacitor filters,
- select the appropriate D/A and A/D converters for a specific application,
- design analogue front ends, and
- calculate SNR and ENOB of an analogue signal processing chain.

Digital Circuit Design 3 SWS/5 ECTS

Teaching Content

- Advanced hardware description language syntax and synthesizable HDL programming
- Practical circuit design and simulation (e.g., arithmetic circuits, finite state machine and interfaces such as SPI, I2C, I2S or UART)
- Verification techniques for digital circuits (e.g., simulation, assertions)
- Workflow for programmable logic devices (FPGAs) and their differences to digital integrated circuit designs (ASICs)
- Synthesis and static timing analysis
- On chip-bus systems (e.g.: Wishbone, AXI-4, APB, AHP)

Competence Acquisition

After finishing this course, students can

- can develop digital circuits using a hardware description language,
- can verify digital circuits using test benches for verification,
- can make hardware circuits configurable via bus addressable control registers,
- can synthesize digital circuits using a tool chain for a FPGA and can deploy a generated design on an evaluation board,
- can design digital communication interfaces (e.g., UART, I^2C or SPI) and can establish the communication between an FPGA and peripheral components, and
- can arithmetically process data received from (or sent to) those peripheral components.

Electronic Packaging 2 SWS/3 ECTS

Teaching Content

- Introduction: overview, interaction between electrical and thermal performance
- Semiconductor manufacturing: backend processes, assembly, and packaging of chips
- Advanced packaging: heterogeneous and multi-component packages, embedding and related concepts
- High integration for PCBs: multi-layer PCB design, signal integrity, routing
- Active and passive cooling concepts for electronic devices, heat sinks, heat pipes
- Chip packaging concepts: flip-chip, wire bonding and more, chip-to-package interactions and thermal considerations
- Advanced 3D PCB design with CAD Tools
- Assigning fabrication and assembly test points for mass production, flying-probe devices, and bed-of-nails test fixtures
- Oscilloscope probe test point design
- Impedance-matched waveguides, differential lines and other copper structures, signal integrity in high-frequency applications
- Interfaces for microwave signal measurement, signal conditioning

Competence Acquisition

After finishing this course, students can

- describe the backend process of a semiconductor manufacturing,
- characterize and compare electronic packages related to their electrical and thermal performance,
- explain advanced packaging concepts,
- explain high integration concepts for PCBs,
- select a proper concept for active and passive cooling purposes,
- describe chip packaging concepts,
- start a project with a state-of-the-art CAD tool to design a highly sophisticated 3D-PCB,
- assign fabrication and assembly test points for automated test methods,
- create and implement test points for oscilloscope probes for prototype commissioning, and
- calculate impedance-matched waveguides, differential lines and other copper structures and implement an interface for measuring signals in the microwave domain.

Design Tools and Laboratory Engineering 3 SWS/3 ECTS

Teaching Content

- Procedural programming and the required programming environment
- Use of debugging techniques for analysis and troubleshooting in software systems
- Introduction to object-oriented programming concepts by designing and implementing classes using practical examples
- Introduction to laboratory operation
- Dimensioning and building of transistor and operational amplifier circuits on prototype boards
- Using lab equipment: power supplies and decoupling ICs, signal generation, multimeters, LCR bridge, oscilloscopes
- Iterative, manual creation of UI-curves and Bode diagrams
- Spice introduction to basic transistor and operational amplifier circuits
- Scientific data processing with numerical analysis tools and graphical presentation of results with respect to guidelines for reports
- PCB design workflow: capturing a schematic, routing of a PCB layout and the automatic generation of manufacturing and assembly data

Competence Acquisition

After finishing this course, students can

- deal with the given development environment (including debugging),
- design and implement procedural software systems,
- design and implement software systems in the form of simple classes,
- set up simple basic circuits on the plug-in board / breadboard / prototype board and put them into operation,
- check electrical characteristics and absolute maximum ratings from data sheets of components by suitable measuring methods,
- analyse networks in the time and frequency domain by means of simulation and in the laboratory,
- independently find errors in circuits and correct them,
- present measurement results and simulation results scientifically and compare them with each other,
- select and apply the appropriate measuring method on state-of-the-art measuring equipment,
- describe the advantages and disadvantages of simulations and can use them as a tool for independent result processing,

- record characteristic curves in the time and frequency domain and present results scientifically,
- check circuit diagrams for functionality in a CAD tool,
- explain how to transfer circuits from a schematic into a printed circuit board,
- generate the documentation of a printed circuit board automatically,
- distinguish what type of data and documentation is necessary to have circuits manufactured/assembled, either as a prototype or as a product developed for mass production,
- analyse existing components, consisting of symbol and footprint, and create them themselves if necessary, and
- integrate and use various online and offline libraries of a CAD tool.

Software Design and Architectures 3 SWS/5 ECTS

Teaching Content

- Object-Oriented Programming: Reinforcement of object-oriented programming concepts, with a focus on practical applications using examples from data structures and algorithms. Examination of fundamental configuration management methodologies, including source code versioning, build automation, and test automation.
- Object-Oriented Modelling: The fundamental principles of object-oriented modelling will be revisited, with a specific focus on Unified Modelling Language (UML) diagrams like package, class, object, sequence, state machine, and activity diagram.
- Object-Oriented Design: The essential principles of class and package design will be explored through hands-on examples to emphasize their significance in creating effective software solutions.
- Additionally, a variety of design patterns will be delved into, with particular attention to their relevance and implementation in embedded systems.

Competence Acquisition

After finishing this course, students can

- implement selected data structures and algorithms in object-oriented programming languages,
- work efficiently with source code versioning systems, build tools and testing frameworks.
- model software system in the form of UML diagrams,
- apply the common design principles for designing classes and packages, and
- select and implement a suitable design pattern for a given design problem.

Data Analysis 3 SWS/5 ECTS

Teaching Content

- Summarizing, organizing, and presenting a set of data using descriptive statistics methods such as frequencies, visualization tools, and correlation and covariance matrix
- Describing and predicting outcomes in uncertain situations using probability theory and their limit theorems
- Definition and applications of random variables including limit theorems, distributions, and model and parameter estimations
- Use of software for statistical analysis

Competence Acquisition

After finishing this course, students can

- calculate and interpret frequencies in a given data set,
- generate and analyse various visualizations to understand the distribution, trend, and outliers in a data set,
- compute and interpret correlation and covariance matrices to understand relationships between variables in a dataset,
- apply probability theory principles to real-life scenarios,
- calculating the likelihood of various outcomes,
- explain and apply the concepts of limit theorems,
- define random variables and provide relevant examples,
- apply limit theorems to random variables and explain the practical implications of these theorems,
- identify the appropriate distributions, formulate models, and estimate their parameters and
- identify and effectively use software tools for data manipulation, numerical computation, statistical modelling and data visualization.

System Requirements Engineering 2 SWS/3 ECTS

Teaching Content

- Introduction into product development basics (development process, project organization, …)
- Definition systems engineering
- V-Model
- Systems engineering activities & work products
- Structuring principles incl. level structure
- Requirements engineering (development vs. management)
- System design within product development - requirements engineering & system
- architecture specification
- Model based systems engineering (MBSE) / SysML
- Basic requirements engineering and applied V-model in the semiconductor industry
- Requirements management & Design-to-Test
- Conformity & compliance
- Integration incl. integration management & test planning within product development
- Requirements management tools
- MBSE / SysML modelling tools

Competence Acquisition

After finishing this course, students can

- can explain the basics about the development of systems in an industrial environment,
- associate the main terms and principles in product development (e.g. what is a development process, what are development methods, meaning of quality gates etc.),
- describe the principles of systems engineering incl. the methods of requirements engineering and system specification,
- understand the meaning and necessity of requirements, use cases, functions, product structure, system architecture and interpret their conformity, including to develop and write requirements and specifications, and
- know common tools applied within systems engineering.

Intercultural Communication 1 SWS/1 ECTS

Teaching Content

- Definition and differentiation of the term “culture”
- Dealing with one's own cultural identity as a starting point for cultural differences, getting to know the own culture better in order to be able to recognize different cultural behaviours
- Cultural diversity – comparison between the culture of the group members
- Fundamentals of communication: theoretical models and practical examples
- Interpersonal communication and communication theory in an intercultural context
- Introduction to group dynamics in an intercultural context
- Communicating effectively with multicultural groups of people in different situations
- Intercultural communication and everyday study life
- Introduction to the study of electronic engineering and the associated rights and obligations as a basis for an intercultural exchange about personal expectations and views on the common everyday study life

Competence Acquisition

After finishing this course, students can

- can define their own cultural identity and cultural patterns and recognize the problems that arise due to cultural differences,
- master selected theoretical models of intercultural communication,
- can contrast mechanisms of human communication (verbal, non-verbal) in an intercultural context,
- can communicate adequately in international teams,
- assess group dynamic effects and their impact on meetings, especially in an international context, and
- know the degree program and the associated rights and obligations and understand the Austrian study culture.