Embedded Systems Engineering

2. Semester

Subject to Approval by the Relevant Bodies

Course	Type	THW	ECTS
Radio Frequency Engineering	IC	4	7
Machine Learning and Optimization	IC	3	6
Computer Architectures and Programming	IC	4	6
Digital Signal Processing	IC	3	5
Software Quality and Security	IC	4	6
		18	**30**

IC Integrated course

THW Hours per week

ECTS European Credit Transfer and Accumulation System

Radio Frequency Engineering 4 SWS/7 ECTS

Teaching Content

- Introduction: Electrodynamics, Maxwell's equations
- Transmission line theory: telegraph equation, time-independent wave-equation, voltage- and current-waves on lossless transmission lines, characteristic impedance, wavelength, phase velocity, reflection coefficient, VSWR, matching, Smith chart, S-parameters microstrip lines, impedance matching networks
- Electromagnetic waves: plane waves in vacuum, plane waves in linear media (skin effect, eddy current, proximity effect, displacement current), electromagnetic waves in wave guides
- Antennas and antenna arrays: Hertz antenna (properties of the electromagnetic field, far-field), linear radiators, antenna arrays
- RF Components: baluns, directional couplers, power divider, circulator, double balanced mixers

Competence Acquisition

After finishing this course, students can

- an explain the mechanism of wave propagation on conductors,
- are able to calculate characteristic impedance, reflection factor and impedance of conducted waves,
- can explain the effects in the quasi-stationary field (skin and proximity effect) and can calculate the penetration depth and the increase in resistance,
- can explain the mechanism of the propagation of electromagnetic waves,
- can calculate the characteristic impedance, reflection factor and velocity of electromagnetic waves in free space and in media,
- can explain the operation principles of RF components and their application,
- can explain the basic operation of antennas,
- can design matching networks, and
- can operate high-frequency measuring devices and carry out measurements with them.

Machine Learning and Optimization 3 SWS/6 ECTS

Teaching Content

- General concepts, algorithms, and models in machine learning such as constrained and unconstrained optimization, cross-validation, and model overfitting and underfitting.
- Data preparation techniques such as handling missing values and outliers, transforming data, binning, and encoding categorical variables
- Selection and application of appropriate machine learning models for clustering, classification, and regression
- Metrics-dependent model evaluation and model deployment into existing systems or processes

Competence Acquisition

After finishing this course, students can

- identify and articulate the key steps in the machine learning process and address their associated challenges,
- apply key algorithms for both constrained and unconstrained optimization to various problem sets,
- implement techniques to prevent overfitting and underfitting in machine learning models,
- select and use appropriate clustering and classification methods to categorize data effectively, and
- construct both shallow and deep neural networks and apply them to real-world data sets for predictive modelling, given appropriate Python knowledge and practice.

Computer Architectures and Programming 4 SWS/6 ECTS

Teaching Content

- Exploration of an example microarchitecture and its Instruction set
- Assembly programming on the example microarchitecture and instruction emulation
- Methods and concepts for improving the performance of microarchitectures (e.g. pipelining, multiprocessors, branch prediction, out-of-order execution, speculative execution, cache memories)
- Workflow for implementing a microarchitecture on programmable system-on-chip or programmable logic devices as well as embedded software development for such hardware architectures
- Benchmarking of a state-of-the-art processor and its architectural features
- Operating system specific hardware components and concepts (memory management, cache coherency, simultaneous multithreading)

Competence Acquisition

After finishing this course, students can

- can explain the working principle of a general-purpose computer and its instruction set,
- will be able to design a simple computer from scratch and program it in assembly language,
- can explain the principles employed in current mobile and desktop computer architectures,
- can assemble a computer architecture on a programmable system on chip or a programmable logic device,
- can program, analyse, and debug software for the designed computer architecture, and
- can describe the fundamental hardware requirements for operating systems.

Digital Signal Processing 3 SWS/5 ECTS

Teaching Content

- Review of discrete-time signals and systems, including signal representation, discrete-time convolution, eigen-signals, the discrete-time Fourier transform (DTFT), and the z-transform
- Analysis of the sampling theorem for sampling and reconstructing signals
- Analysis of the relationships between the continuous-time Fourier transform (CTFT), the discrete-time Fourier transform (DTFT), the discrete Fourier transform (DFT) and the fast Fourier transform (FFT)
- Design of finite impulse response (FIR), infinite impulse response (IIR), and adaptive filters to attenuate, equalize and generate signals
- Design of multi-rate systems to implement sample rate conversion and polyphase filters

Competence Acquisition

After finishing this course, students can

- explain the implications of finite signal and finite word length,
- apply sampling and quantization theory to the analysis of continuous-time signals in the digital domain,
- explain digital filters, including FIR, IIR, adaptive and multirate filters,
- design digital filters using software tools and implement them on a digital signal processor, and
- implement signal transformations such as the Fast Fourier Transform on a digital signal processor.

Software Quality and Security 4 SWS/6 ECTS

Teaching Content

- Software Inspection: This process includes examining source code to identify "Bad Smells", applying software metrics for code quality assessment, and using static code analysis to detect potential bugs
- Software testing: It encompasses unit tests, which target individual software components, and integration tests, which validate interactions between components to ensure seamless functionality
- Secure coding: This includes programming techniques like encapsulation, input validation, output encoding, secure data representation, and robust error handling & logging
- Cryptography in practice: It focuses on cryptographic techniques to secure data at rest, protecting sensitive information from unauthorized access and ensuring data integrity
- Reverse engineering: This involves analysis of binaries to understand their structure and functionality which is commonly used in debugging and security research

Competence Acquisition

After finishing this course, students can

- perform manual and tool-based inspections of software systems to locate errors and code smells,
- design and implement unit- and integration test cases,
- master the most important techniques of secure coding and can therefore identify and remove security gaps in the source code,
- apply and configure the selected cryptographic algorithms, and
- master the technique of reverse engineering binary files to check for vulnerabilities or malware.