

Technology
Arts Sciences
TH Köln

Fakultät 07 für Informations-, Medien- und Elektrotechnik

Bachelor Technische Informatik 2020

Modulhandbuch

Version: 3.10.2025-08-25-12-18-36

Die neueste Version dieses Modulhandbuchs ist verfügbar unter:

<https://f07-studieninfo.web.th-koeln.de/mhb/current/de/BaTIN2020.html>

1. Studiengangsbeschreibung

Der Bachelor-Studiengang "Technische Informatik" vermittelt fundierte Kenntnisse in der Entwicklung und Anwendung komplexer informationstechnischer Systeme. Die Studierenden lernen, Software und Hardware effektiv zu kombinieren, um innovative Lösungen für aktuelle technologische Herausforderungen zu schaffen. Denkbar ist hier beispielsweise die Entwicklung eines Systems, das Sensordaten in einem Smart Home verarbeitet und analysiert. Informatik und Systems-Engineering sind Schlüsselkompetenzen des 21. Jahrhunderts.

Ausrichtung des Studiengangs

Der Studiengang ist ein Informatik-Studiengang mit einer Regelstudienzeit von sieben Semestern. Er bietet ein breites Themenspektrum, aus dem die Studierenden einem von sechs Studienschwerpunkten wählen können.

Ergänzend werden Module aus den Bereichen Nachrichtentechnik, Akustik und Elektrotechnik angeboten.

Berufsfelder

Die Absolventinnen und Absolventen dieses Studiengangs finden spannende Berufsmöglichkeiten in zahlreichen technologischen Bereichen. In der Softwareentwicklung sind sie gefragt, wo sie maßgeschneiderte Anwendungen und Systeme für die unterschiedlichsten Branchen entwickeln. In der IT-Branche und bei der Entwicklung intelligenter Technologien in Bereichen wie der Medizintechnik oder der Energietechnik finden Fachkräfte für eingebettete Systeme vielfältige Einsatzmöglichkeiten. Absolventinnen und Absolventen im Bereich der Künstlichen Intelligenz sind gefragte Fachkräfte, die Algorithmen für maschinelles Lernen und datenbasierte Entscheidungsfindung entwickeln. In der Telekommunikation setzen sie ihr Wissen über Netzwerke und Datenübertragung ein, um die Effizienz und Sicherheit von Kommunikationssystemen zu verbessern.

Die im Studiengang vermittelten Kompetenzen machen unsere Absolventinnen und Absolventen zu wertvollen Teammitgliedern in Start-ups ebenso wie in etablierten Unternehmen.

Studienverlauf

Zu Beginn des Studiums liegt der Schwerpunkt auf den Grundlagen der Informatik in den Bereichen Software und Hardware sowie der Mathematik, wobei praktische Übungen und Prüfungen das theoretische Wissen festigen. In den folgenden Semestern erweitern sich die Themen auf fortgeschrittene Inhalte wie Betriebssysteme, Netzwerke, IT-Sicherheit und Software Engineering, wobei ein besonderer Fokus auf der systematischen Herangehensweise an die Entwicklung und Optimierung von technischen Systemen liegt. Zahlreiche Wahlmodule bieten die Möglichkeit, sich auf spezielle Interessen zu konzentrieren. Praktische Fähigkeiten werden durch Projektarbeiten und Praktika intensiv gefördert. Das Studium schließt mit einer Praxisphase und der Bachelor-Arbeit ab. Dieser strukturierte Ablauf vermittelt den Studierenden sowohl theoretisches Wissen als auch praktische Erfahrungen mit besonderem Schwerpunkt auf der anwendungsorientierten Integration verschiedener Technologien, die für den Erfolg in IT- und techniknahen Berufen unerlässlich sind.

Studienbegleitend wird die professionelle Weiterbildung zum Cisco Certified Network Associate (CCNA) angeboten.

Auslandsstudium

Wir unterstützen unsere Studierenden aktiv bei der Integration eines Studienaufenthalts an einer ausländischen Hochschule. Die dort erbrachten Leistungen werden für das deutsche Studium anerkannt. Hierzu bestehen enge Kontakte zu Hochschulen in mehreren europäischen und außereuropäischen Ländern. Das fünfte Studiensemester kann an einer dieser Hochschulen absolviert werden.

Studienvoraussetzungen

Fachhochschulreife (schulischer und praktischer Teil) oder Abitur bzw. gleichwertiger Abschluss.

2. AbsolventInnenprofil

AbsolventInnen des Studiengangs B. Sc. Technische Informatik verfügen über fundierte Kenntnisse in Informatik, Softwareentwicklung und technischer Systemintegration. Sie sind in der Lage, komplexe informationstechnische Systeme zu konzipieren, zu entwickeln, zu analysieren und zu betreiben. Ihr interdisziplinäres Wissen, gepaart mit praxisnaher Projektarbeit, qualifiziert sie für vielfältige Aufgaben in der Software- und Systementwicklung, je nach Wahl des Schwerpunktes insbesondere in den Bereichen eingebetteter Systeme, verteilter Anwendungen und der Künstlichen Intelligenz.

Ziel des Bachelorstudiengangs Technische Informatik ist es, die Studierenden zu reflektierten Fachkräften auszubilden, die in der Lage sind, moderne informationstechnische Systeme in ihrer gesamten Breite zu entwerfen, zu realisieren und weiterzuentwickeln -- von der Hardware-nahen Programmierung über systemnahe Software bis hin zu vernetzten, intelligenten Anwendungen.

Die AbsolventInnen erwerben grundlegende und fortgeschrittene Kompetenzen in der Informatik, Mathematik, Kommunikationstechnik und Elektrotechnik. Sie können dabei einen der drei Studienschwerpunkte wählen:

- Eingebettete Systeme
- Internet of Things
- Netze und Verteilte Systeme
- Smart Systems
- Software-Systeme

Durch projektorientierte Studienanteile, praxisnahe Lehrformate und eine vertiefte Auseinandersetzung mit realen Anwendungsfeldern werden methodische, technische und soziale Kompetenzen gleichermaßen gefördert. Die AbsolventInnen sind mit modernen Entwicklungsmethoden, agilen Prozessen und interdisziplinärer Teamarbeit vertraut.

Die AbsolventInnen des Bachelor-Studiengangs Technische Informatik entwickeln in folgenden Bereichen ihr individuelles Profil:

- Sie sind in der Lage, komplexe Soft- und Hardwaresysteme zu analysieren, modellieren und umsetzungsreif zu gestalten -- insbesondere in sicherheitskritischen und verteilten Systemumgebungen.
- Sie verfügen über anwendungsorientierte Kompetenzen zur Programmierung, Systemspezifikation, Simulation, Verifikation und Qualitätssicherung von IT-Systemen.
- Sie beherrschen die Konzeption und den Betrieb verteilter Systeme, Cloud-Anwendungen, Datenbanken, eingebetteter Systeme und Komponenten der Künstlichen Intelligenz.
- Über die projektbasierte Ausbildung lernen sie, sich schnell in neue Technologien einzuarbeiten und diese im Team effizient in die Praxis umzusetzen -- auch in interkulturellen Kontexten.
- Sie sind befähigt, gesellschaftliche, ethische und rechtliche Fragestellungen zu reflektieren und ihre technische Arbeit nachhaltig und verantwortungsvoll zu gestalten.
- Mit ihren kommunikativen, interdisziplinären und organisatorischen Fähigkeiten können sie sowohl in Start-ups als auch in Großunternehmen technische Verantwortung übernehmen und Führungsrollen anstreben.

Der Studiengang qualifiziert sowohl für den direkten Berufseinstieg in Bereichen wie Softwareentwicklung, Embedded Systems, Automatisierungstechnik, Telekommunikation und Data Science als auch für ein anschließendes Masterstudium. Sie können aber auch im technischen Vertrieb, in der Schulung, in der technischen Kundenberatung und im Projektmanagement (bspw. als Product-Owner oder Scrum-Master) tätig werden. InformatikerInnen sind mit diesen Tätigkeiten in vielen Branchen im Einsatz: bei Software-Herstellern, Entwicklungsunternehmen, in der Informations- und Kommunikationstechnik, bei Beratungsfirmen oder Behörden.

3. Handlungsfelder

Zentrale Handlungsfelder im Studium sind Entwicklung und Design, Forschung und Innovation, Leitung und Management sowie Qualitätssicherung und Tests. Die Profil-Modulmatrix stellt dar, welche Handlungsfelder durch welche Module adressiert werden.

Forschung und Entwicklung

In diesen Bereich fallen das Erforschen und Entwickeln von neuen Technologien, Algorithmen, Verfahren, Geräten, Komponenten und Anlagen. Das umfasst sowohl Grundlagen- und Industrieforschung als auch die spezialisiertere Entwicklung wie in der Medientechnologie, Optometrie, Informationstechnik und Elektrotechnik sowie Informatik und Systems-Engineering.

System- und Prozessmanagement

Hierunter fällt die Planung, Konzeption, Überwachung, Betrieb und Instandhaltung von Systemen und Prozessen. Dies beinhaltet auch das Management von Produktionsprozessen, die Qualitätssicherung und die Koordination von Arbeitsgruppen sowie die IT-Administration und das Projektmanagement.

Innovation und Anwendung

Innovation und Anwendung umfasst die Auslegung, Entwicklung und Nutzung innovativer Anwendungen und Systeme in technischen Disziplinen. Dazu gehört auch die Erstellung und Gestaltung von Medieninhalten und -produkten, die Entwicklung elektronischer, informatischer, medientechnologischer, akustischer oder optischer Komponenten und Systeme sowie die Integration von informationstechnischen Lösungen in technischen Anwendungen.

Analyse, Bewertung und Qualitätssicherung

Die Analyse und Bewertung von Verfahren, Systemen, Algorithmen und Geräten zur Sicherung der Qualität von Produkten und Prozessen, beinhaltet die Reflexion und Bewertung von medialen Inhalten und klinischen Studien sowie die Untersuchung visueller und akustischer Wahrnehmungsprozesse.

Interaktion und Kommunikation

Die Fähigkeit zu interdisziplinärer Zusammenarbeit und Vermittlung zwischen gestalterisch Tätigen, technischen Akteuren, Auftraggebern und Anwendern. Betont die Bedeutung von Soft-Skills wie Teamarbeit und Präsentationsfähigkeiten in technischen Berufsfeldern.

4. Kompetenzen

Die Module des Studiengangs bilden Studierende in unterschiedlichen Kompetenzen aus, die im Folgenden beschrieben werden. Die Profil-Modulmatrix stellt dar, welche Kompetenzen durch welche Module adressiert werden.

Systemdenken und Abgrenzung von Systemgrenzen

Verstehen und Identifizieren der Grenzen verschiedener Systeme, einschließlich der Abgrenzung relevanter Aspekte von externen, unbeflussbaren Faktoren.

Abstraktion und Modellierung

Fähigkeit zur Vereinfachung und Verallgemeinerung von komplexen Problemen, Entwicklung und Bewertung unterschiedlicher Modelle über verschiedene Fachdisziplinen hinweg.

Analyse natürlicher und technischer Phänomene

Identifikation, Benennung und Erklärung relevanter Phänomene in realen Szenarien, unter Einbeziehung naturwissenschaftlicher Grundlagen und technischer Zusammenhänge.

MINT-Kompetenz

Kenntnis und Anwendung von Modellen und Prinzipien aus Mathematik, Informatik, Naturwissenschaften und Technik für die Problemlösung.

Simulation und Analyse technischer Systeme

Einsatz von Software und Werkzeugen zur Simulation und Analyse technischer Systeme, einschließlich der Entwicklung von Simulationsmodellen.

Entwurf und Realisierung von Systemen und Prozessen

Gestaltung und Implementierung von technischen Lösungen und Prozessen, unter Berücksichtigung technischer, ökonomischer und ökologischer Standards und Prinzipien.

Prüfen und Bewerten von Systemen und Prozessen

Durchführung von Tests samt Verifikation und Validierung, um die Einhaltung von Standards und die Funktionalität von Systemen und wirtschaftlicher Aspekte von Prozessen zu gewährleisten.

Informationsbeschaffung und -auswertung

Fähigkeit zur systematischen Recherche, Analyse und Bewertung von Informationen unter Einbeziehung relevanter Kontexte.

Kommunikation und Präsentation

Effektive Darstellung und Erläuterung komplexer technischer Inhalte an unterschiedliche Zielgruppen in deutscher und englischer Sprache.

Betriebswirtschaftliches und rechtliches Wissen

Anwendung von Grundkenntnissen in Betriebswirtschaft und Recht bezogen auf technische und gestalterische Projekte und Entscheidungen.

Teamarbeit und interdisziplinäre Zusammenarbeit

Fähigkeit zur Arbeit in Teams, einschließlich der effektiven Kommunikation und Kooperation mit Fachvertretern anderer Disziplinen.

Entscheidungsfindung in unsicheren Situationen

Strategische Entscheidungsfindung basierend auf fachlich fundierten Analysen, selbst unter Unsicherheit.

Berücksichtigung gesellschaftlicher und ethischer Werte

Integration von ethischen und gesellschaftlichen Werten bei der Gestaltung von Systemen und Medien und Reflexion beruflichen Handelns.

Lernkompetenz und Adaptionfähigkeit

Motivation und Fähigkeit zum lebenslangen Lernen sowie zur Anpassung an technologische und methodische Neuerungen.

Selbstorganisation und Selbstreflexion

Kompetenz in der Selbstorganisation beruflicher und lernbezogener Aufgaben sowie kritische Reflexion des eigenen Handelns.

Kommunikative und interkulturelle Kompetenzen

Effektive Kommunikation und Zusammenarbeit in interkulturellen und internationalen Kontexten sowie mediale Kompetenzen.

Spezifische Fachkenntnisse und Fertigkeiten

Vertiefte Kenntnisse und Fertigkeiten, die auf die Anforderungen und Besonderheiten der einzelnen Fachgebiete wie Medientechnologie, Optometrie, Informationstechnik und Elektrotechnik sowie Informatik und Systems-Engineering.

5. Studienverlaufspläne

Im Folgenden sind studierbare Studienverlaufspläne dargestellt. Andere Studienverläufe sind ebenso möglich. Beachten Sie bei Ihrer Planung dabei jedoch, dass jedes Modul in der Regel nur einmal im Jahr angeboten wird. Beachten Sie auch, dass in einem bestimmten Semester und Wahlbereich ggf. mehrer Module gewählt werden müssen, um die dargestellte Summe an ECTS-Kreditpunkten zu erlangen.

5.1 Studienverlaufsplan

Sem.	Kürzel	Modulbezeichnung	Pflicht (PF) Wahl- bereich (WB)	ECTS
1	DR	Digitalrechner	PF	5
	EG	Elektrotechnische Grundlagen für die Technische Informatik	PF	5
	MA1	Mathematik 1	PF	10
	PI1	Praktische Informatik 1	PF	5
	PP	Programmierpraktikum	PF	5
2	FSA	Formale Sprachen und Automatentheorie	PF	5
	AD	Algorithmen und Datenstrukturen	PF	5
	MA2	Mathematik 2	PF	10
	PI2	Praktische Informatik 2	PF	5
	GSP	Grundlagen der Systemprogrammierung	PF	5
3	BVS1	Betriebssysteme	PF	6
	GUI	Graphische Oberflächen und Interaktion	PF	5
	DB1	Datenbanken 1	PF	5
	NP	Netze und Protokolle	PF	5
	SE	Software Engineering	PF	5
	SIG	Signalverarbeitung	PF	5
4	BVS2	Verteilte Systeme	PF	5
	BWR	Betriebswirtschaft und Recht	PF	5
	ITS	IT-Sicherheit	PF	5
	SWP	Softwarepraktikum	PF	6
	WM	Wahlbereich Informatik und Systems-Engineering	WB	10
5	SYP	Systementwicklungs-Projekt	PF	7
	PUK	Präsentation und Kommunikation	PF	3
	XIB	Fachübergreifende Kompetenzen und Soft-Skills	PF	5
	WM	Wahlbereich Informatik und Systems-Engineering	WB	15
6	IPM	IT-Projektmanagement	PF	5
	PRA	Praxisphase	PF	8
	WM	Wahlbereich Informatik und Systems-Engineering	WB	15
7	PRA	Praxisphase	PF	15
	BAA	Bachelorarbeit	PF	12
	KOLL	Kolloquium zur Bachelorarbeit	PF	3

5.2 Alternativer Studienverlaufsplan (verminderter Workload)

Sem.	Kürzel	Modulbezeichnung	Pflicht (PF) Wahl- bereich (WB)	ECTS
1	MA1	Mathematik 1	PF	10
	PI1	Praktische Informatik 1	PF	5
	PP	Programmierpraktikum	PF	5
2	MA2	Mathematik 2	PF	10
	PI2	Praktische Informatik 2	PF	5
	GSP	Grundlagen der Systemprogrammierung	PF	5
3	BVS1	Betriebssysteme	PF	6
	GUI	Graphische Oberflächen und Interaktion	PF	5
	DR	Digitalrechner	PF	5
	EG	Elektrotechnische Grundlagen für die Technische Informatik	PF	5
4	BVS2	Verteilte Systeme	PF	5
	FSA	Formale Sprachen und Automatentheorie	PF	5
	ITS	IT-Sicherheit	PF	5
	AD	Algorithmen und Datenstrukturen	PF	5
5	DB1	Datenbanken 1	PF	5
	SE	Software Engineering	PF	5
	SIG	Signalverarbeitung	PF	5
	NP	Netze und Protokolle	PF	5
6	SWP	Softwarepraktikum	PF	6
	BWR	Betriebswirtschaft und Recht	PF	5
	WM	Wahlbereich Informatik und Systems-Engineering	WB	10
7	SYP	Systementwicklungs-Projekt	PF	7
	PUK	Präsentation und Kommunikation	PF	3
	XIB1	Fachübergreifende Kompetenzen und Soft Skills 1	WB	5
	WM	Wahlbereich Informatik und Systems-Engineering	WB	5
8	IPM	IT-Projektmanagement	PF	5
	WM	Wahlbereich Informatik und Systems-Engineering	WB	15
9	PRA	Praxisphase	PF	8
	WM	Wahlbereich Informatik und Systems-Engineering	WB	10
10	PRA	Praxisphase	PF	15
	BAA	Bachelorarbeit	PF	12
	KOLL	Kolloquium zur Bachelorarbeit	PF	3

6. Module

Im Folgenden werden die Module des Studiengangs in alphabetischer Reihenfolge beschrieben.

6.1 AD - Algorithmen und Datenstrukturen

Modulkürzel	AD_BaTIN2020
Modulbezeichnung	Algorithmen und Datenstrukturen
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	AD - Algorithmen und Datenstrukturen
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	2
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Pascal Cerfontaine/Professor Fakultät IME
Dozierende*r	Prof. Dr. Pascal Cerfontaine/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt Kenntnisse zum Umgang mit Algorithmen und Datenstrukturen sowie ein tieferes Verständnis der zugrundeliegenden Konzepte und Techniken. Im Fokus dieses steht die exemplarische Nutzung der Programmierung in C++ zur Realisierung der behandelten Datenstrukturen und Algorithmen. In praktischer Arbeit analysieren die Studierenden Problemstellungen im Systemumfeld (K1, K2, K4), implementieren Lösungen auf der Grundlage anerkannter Konzepte und Methoden (K3) mit Hilfe von Standardwerkzeugen (K6, K9) und prüfen sie (K7). Sie recherchieren dazu in Dokumentationen (K8, K15) und passen vorhandene Software an (K10).

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. In den Übungen und insbesondere im Praktikum arbeiten die Studierenden in Kleingruppen und verteidigen ihre Lösungen (K8, K13, K16).

Wozu: Die Verwendung und Beurteilung von Standarddatenstrukturen und Standardalgorithmen ist essentiell für heutige komplexe Softwaresysteme. Entsprechende Programmierkenntnisse und Wissen über die zugehörigen Grundlagen sind somit unverzichtbar für die Erstellung moderner Software (HF1). Durch ihre praktische Programmierarbeit erwerben die Studierenden Erfahrungen, die wichtig sind für die Erfassung von Anforderungen, die Entwicklung von Konzepten zur technischen Lösung und zu ihrer Bewertung (HF2) sowie zur Ausführung von Programmen (HF3). Die Durchführung im Team mit dem Dozenten als "Auftraggeber" stärkt die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte**Vorlesung / Übungen**

Konzepte der objektorientierten Programmierung in C++
 grundlegende lineare und hierarchische Datenstrukturen in der Programmierung
 Lineare Datenstrukturen (z.B. lineare Listen, verkettete Listen usw.)
 Hierarchische Datenstrukturen (z.B. k-näre Bäume, binäre Suchbäume usw.)
 Analyse der Komplexität von Algorithmen
 Wichtige Suchalgorithmen
 Funktionsweise wesentlicher Sortialgorithmen
 Konzepte der objektorientierten Programmierung in C++ anwenden
 Qualität von Datenstrukturen und Algorithmen einschätzen
 Programmieren von Such- und Sortierv Verfahren

Praktikum

Datenstrukturen und Algorithmen für spezielle Anwendungen selbständig entwerfen
 lineare und hierarchische Datenstrukturen in C++ implementieren
 Sortialgorithmen in C++ umsetzen
 Suchalgorithmen in C++ realisieren
 Anwendung der unter "Kenntnisse (fachliche Inhalte)" genannten Aspekte auf praxisbezogene Szenarien durch selbstständige Arbeit in kleinem Team.

Lehr- und Lernmethoden

- Vorlesung / Übungen
- Praktikum

Prüfungsformen mit Gewichtung siehe Prüfungsordnung

Workload 150 Stunden

Präsenzzeit 45 Stunden $\pm$ 4 SWS

Selbststudium 105 Stunden

Empfohlene Voraussetzungen

- Modul PI1: Sicherer Umgang mit einer Programmiersprache.
- Grundlagen der Programmierung in einer höheren Programmiersprache

Zwingende Voraussetzungen Praktikum erfordert Anwesenheit im Umfang von: 3 Praktikumstermine

Empfohlene Literatur

- Sedgewick, Robert: Algorithmen in C
- Ottmann, Widmayer: Algorithmen und Datenstrukturen
- Heun: Grundlegende Algorithmen
- Wirth, Niklaus: Algorithmen und Datenstrukturen
- Elektronische Verweise auf ebooks und Online Tutorials

Verwendung des Moduls in weiteren Studiengängen AD in Bachelor Informatik und Systems-Engineering 2024

Besonderheiten und Hinweise

Letzte Aktualisierung 19.7.2025, 14:32:16

6.2 ASN - Angewandte Statistik und Numerik

Modulkürzel	ASN_BaTIN2020
Modulbezeichnung	Angewandte Statistik und Numerik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	ASN - Angewandte Statistik und Numerik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Holger Weigand/Professor Fakultät IME
Dozierende*r	Prof. Dr. Holger Weigand/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt die Kompetenz, mathematische Modelle zur Beschreibung technischer Systeme zu entwerfen (K2, K5, K11), diese effizient zu implementieren und ihre Grenzen zu benennen (K1, K19). Der Studierende kann Informationen mathematisch aus- und bewerten (K12).

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs/Übungsteil und betreut parallel dazu ein Praktikum, in dem die Studierenden bekannte und selbst entwickelte Algorithmen implementieren.

Wozu: Die erworbenen Kompetenzen unterstützen den Studierenden bei der Entwicklung von Algorithmen für die Forschung (HF 1). Er kann die Güte von Algorithmen bei größeren technischer Systemen abschätzen bzw. sie in solchen Systemen realisieren (HF2). Bei der Planung und Realisierung von Systemen zur Verarbeitung von Informationen für technische Anwendungen (HF3) kann er abstrakte Modelle entwerfen, speziell bei Berechnungssystemen.

Modulinhalte

Vorlesung / Übungen

Rechnerarithmetik
 Fehlerrechnung, Kondition einer Matrix
 Gaußalgorithmus mit Spaltenpivotisierung
 Interpolation
 Nullstellenprobleme (Bisektion, Newton, Varianten von Newton, Fixpunktiteration)
 Iterationsverfahren für lineare GS
 Regressionsanalyse
 Wahrscheinlichkeitsrechnung

Praktikum

Weitergabe von Meßfehlern abschätzen können
 numerische Algorithmen anwenden können
 Trendfunktionen aufstellen können
 mit Wahrscheinlichkeiten umgehen können

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS

Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul MA1: Grundlegende mathematische Kenntnisse, insbesondere Funktionen und Differentialrechnung anwenden ▪ Modul MA2: Methoden der linearen Algebra anwenden können ▪ Modul PI1: Grundbegriffe der Programmierung anwenden ▪ Grundlegende mathematische Kenntnisse, insbesondere Funktionen und Differentialrechnung anwenden ▪ Methoden der linearen Algebra anwenden können ▪ Grundbegriffe der Programmierung anwenden
Zwingende Voraussetzungen	▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Vorlesung / Übungen ▪ Praktikum erfordert Anwesenheit im Umfang von: 4 Termine
Empfohlene Literatur	▪ Knorrenschild: Numerische Mathematik (Fachbuchverlag) ▪ Papula: Mathematik für Ingenieure und Naturwissenschaftler, Band 1+2 (Vieweg)
Verwendung des Moduls in weiteren Studiengängen	▪ AM in Bachelor Elektrotechnik 2020 ▪ ASN in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ ASN in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.3 AT - Antennentechnik

Modulkürzel	AT_BaTIN2020
Modulbezeichnung	Antennentechnik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	AT - Antennentechnik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Rainer Kronberger/Professor Fakultät IME
Dozierende*r	Prof. Dr. Rainer Kronberger/Professor Fakultät IME

Learning Outcome(s)

Die Studierenden können spezielle elektromagnetische Probleme lösen, indem sie hierfür geeignete spezielle Methoden der Hochfrequenztechnik und Elektrotechnik anwenden, um später Antennen für hochfrequente Anlagen, Systeme und Baugruppen zu analysieren, entwickeln und herzustellen

Modulinhalte**Vorlesung**

Elektromagnetische Wellen
 Maxwellsche Gleichungen
 Wellengleichung
 Wellenausbreitung
 Wellen an Grenzflächen
 Reflexion, Beugung und Brechung
 Antennen
 Definition
 Parameter
 Elementarstrahler
 Gruppenantennen
 Flächenstrahler
 Mobilfunkantennen
 Besondere Formen
 Elektromagnetische Wellen verstehen
 Umgang mit Simulationswerkzeugen
 Erlernen von Messverfahren und Vorschriften
 Wirkungsweise von Antennen verstehen

Projekt

Hochfrequenztechnische Messaufbauten verstehen
 HF-Simulationswerkzeuge bedienen
 HF-Antennenmessgeräte fachgerecht einstellen
 Antennenmessungen durchführen
 Antennensimulationen durchführen
 Wissenschaftlichen Bericht verfassen

Lehr- und Lernmethoden

- Vorlesung
- Projekt

Prüfungsformen mit Gewichtung
 siehe Prüfungsordnung

Workload
 150 Stunden

Präsenzzeit
 34 Stunden $\pm$ 3 SWS

Selbststudium
 116 Stunden

Empfohlene Voraussetzungen

- Modul EG: Erweiterte Kenntnisse der Elektrotechnik und Grundlagen der Grundlegende Kenntnisse der Hochfrequenztechnik
- Gleichstromtechnik
- Wechselstromtechnik
- Elektronik
- Grundlagen Mathematik
- Grundlagen der Hochfrequenztechnik

Zwingende Voraussetzungen
 Projekt erfordert Anwesenheit im Umfang von: 6 Termine

Empfohlene Literatur

- Meinke/ Gundlach: Taschenbuch der Hochfrequenztechnik Bd. 1-3 Springer Verlag
- Detlefsen/Siart: Grundlagen der HF-Technik. Oldenbourg Verlag
- Zinke/ Brunswig: Hochfrequenztechnik 1, Filter, Leitungen, Antennen, Springer Verlag
- Kark: Antennen und Strahlungsfelder, Springer Verlag

Verwendung des Moduls in weiteren Studiengängen	▪ AT in Bachelor Elektrotechnik 2020 ▪ AT in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ AT in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise

Letzte Aktualisierung 19.7.2025, 14:32:16

6.4 ATS - Autonome Systeme

Modulkürzel	ATS_BaTIN2020
Modulbezeichnung	Autonome Systeme
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	ATS - Autonome Systeme
ECTS credits	5
Sprache	deutsch und englisch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Chunrong Yuan/Professor Fakultät IME
Dozierende*r	Prof. Dr. Chunrong Yuan/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt Kompetenzen zur Erstellung von autonomen Systemen (AS) in allen relevanten Aspekten und Arbeitsschritten von der Auslegung und Planung des gesamten Systems (K.1, K.4), Auswahl und Bewertung der Komponenten (K.8, K.9), Entwicklung der Software für die Sensordatenverarbeitung und intelligente Robotersteuerung unter der Verwendung von Methoden wie z.B. KI (Künstliche Intelligenz) und Robotersehen (K.2, K.3), die Integration von Software und Hardware Komponenten (K.5, K.6), bis zur Inbetriebnahme und Validierung des gesamten robotischen Systems (K.7, K.10).

Womit: Die Dozentin vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs/Übungsteil und betreut parallel dazu Miniprojekte, in denen die Studierenden ihre Kenntnisse anwenden und relevante Komponenten für AS entwickeln.

Wozu: Kompetenzen in der Entwicklung eines AS sind essentiell für technische Informatiker*innen, die im HF1 arbeiten wollen. Durch das Erlernen und die Anwendung von aktuellen Methoden und Techniken im Bereich KI und Robotik anhand robotischer Plattformen erwerben die Studierenden zudem Erfahrungen, die essentiell für das HF2 sind, u.a. Anforderungen erfassen, Konzepte zur technischen Lösung entwickeln und diese zu bewerten.

Modulinhalte**Vorlesung**

Sensorik
 Bewegungsmesser
 Ausrichtungsmessung
 Position- und Entfernungsmessung
 Kameras und Kameramodelle
 Fortbewegung
 Radfahrzeuge
 Laufmaschinen
 Sensordatenverarbeitung und Merkmalsgewinnung
 Kantendetektion
 Linienextraktion
 Punktdetektor und -deskriptor
 Erkennung und Modellierung
 Objektdetektion
 Ortserkennung
 3D Struktur- und Bewegungsschätzung
 Navigation
 Lokalisierung
 Kartierung
 Wegplanung

Praktikum

Im Team: Entwicklung von Systemen mit intelligenten Verhalten für autonome Sensordatenverarbeitung und echtzeitige Robotersteuerung. Das Ziel der Projekte besteht darin, Prototypen zu entwickeln, die entsprechenden Funktionalitäten nachweisen.

Übungen

Charakterisierung von Sensoren
 Merkmalsgewinnung
 Bildvergleich und Clustering
 Bildbasierte Ortserkennung
 Bewegungsanalyse
 Programmierung der Roboterverhalten

Lehr- und Lernmethoden	▪ Vorlesung ▪ Praktikum ▪ Übungen
-------------------------------	---

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul AD: Kompetenz in der Analyse und Realisierung von Algorithmen ▪ Modul SIG: Kenntnisse in der Signalverarbeitung ▪ Modul SYP: Kompetenz in der Entwicklung von Software und Projekten ▪ Modul ES: Grundkenntnisse in der hardwarenahe Softwareentwicklung ▪ Kompetenz in der Analyse und Realisierung von Algorithmen ▪ Kenntnisse in der Signalverarbeitung und Mathematik ▪ Kompetenz in der Entwicklung von Software und Projekten ▪ Grundkenntnisse in der hardwarenahen Softwareentwicklung
-----------------------------------	--

Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: 1 Termin
Empfohlene Literatur	▪ Hertzberg: Mobile Roboter: Eine Einführung aus Sicht der Informatik, Springer Vieweg, 2012
Verwendung des Moduls in weiteren Studiengängen	▪ ATS in Bachelor Elektrotechnik 2020 ▪ ATS in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ ATS in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.5 BAA - Bachelorarbeit

Modulkürzel	BAA_BaTIN2020
Modulbezeichnung	Bachelorarbeit
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	BAA - Bachelorarbeit
ECTS credits	12
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	7
Häufigkeit des Angebots	jedes Semester
Modul-Verantwortliche*r	Studiengangsleiter(in) Bachelor Technische Informatik / Informatik und Systems-Engineering
Dozierende*r	verschiedene Dozenten*innen / diverse lecturers

Learning Outcome(s)

Was: Das Modul vermittelt die Fähigkeit, eine Aufgabe der Informatik oder eine ingenieurwissenschaftliche Aufgabe selbstständig mit Hilfe wissenschaftlicher Methoden zu bearbeiten.

Womit: Auf Basis einer individuellen Vereinbarung des Studierenden mit einem Dozenten der F07 erhält der Studierende eine qualifizierte Aufgabenstellung aus dem Bereich der technischen Informatik die eigenständig und innerhalb einer beschränkten Frist erfolgreich bearbeitet und dokumentiert werden muss. Die Bachelorarbeit kann auch extern in einem Unternehmen durchgeführt werden.

Wozu: Das Berufsleben eines Informatikers ist geprägt durch die Bearbeitung komplexerer Aufgabenstellung aus dem Bereich der Informatik und/oder der Ingenieurwissenschaften. Hierbei stellt die vollumfängliche Bearbeitung (Recherche, Konzepterstellung, Implementierung, Verifikation, Validierung, Dokumentation) der Aufgabe eine wesentlichen Anforderung dar. Dies wird im Rahmen dieses Moduls quasi als "Einstieg in das Berufsleben" vermittelt.

Modulinhalte

Abschlussarbeit

Die Bachelorarbeit ist eine schriftliche Hausarbeit. Sie soll zeigen, dass die oder der Studierende befähigt ist, innerhalb einer vorgegebenen Frist ein Thema aus ihrem oder seinem Fachgebiet sowohl in seinen fachlichen Einzelheiten als auch in den fachübergreifenden Zusammenhängen nach wissenschaftlichen und fachpraktischen Methoden selbstständig zu bearbeiten. Die interdisziplinäre Zusammenarbeit kann auch bei der Abschlussarbeit berücksichtigt werden.

Lehr- und Lernmethoden	Abschlussarbeit
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	360 Stunden
Präsenzzeit	0 Stunden $\pm$ 0 SWS
Selbststudium	360 Stunden
Empfohlene Voraussetzungen	
Zwingende Voraussetzungen	siehe Prüfungsordnung §26 Abs. 1
Empfohlene Literatur	

Verwendung des Moduls in weiteren Studiengängen	▪ BAA in Bachelor Elektrotechnik 2020 ▪ BAA in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ BAA in Bachelor Medientechnologie 2020 ▪ BAA in Bachelor Medientechnologie 2024 ▪ BAA in Bachelor Optometrie 2021 ▪ BAA in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	Siehe auch Prüfungsordnung §24ff. Kontaktieren Sie frühzeitig einen Professor der Fakultät für die Erstbetreuung der Abschlussarbeit.
Letzte Aktualisierung	19.7.2025, 14:32:16

6.6 BVS1 - Betriebssysteme

Modulkürzel	BVS1_BaTIN2020
Modulbezeichnung	Betriebssysteme
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	BVS1 - Betriebssysteme
ECTS credits	6
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	3
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Studiengangsleiter(in) Bachelor Technische Informatik / Informatik und Systems-Engineering
Dozierende*r	Prof. Dr. Andreas Behrend/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt Kompetenzen zum Umgang mit Betriebssystemen und Diensten in verteilten Systemen sowie ein tieferes Verständnis der zugrundeliegenden Konzepte und Techniken. Im Fokus dieses ersten Moduls (gefolgt von BVS2) steht die Nutzung von Programmier Techniken und -schnittstellen, die eine Systemsoftware typischerweise zur Realisierung nebenläufiger, kooperierender Software im lokalen und verteilten Umfeld anbietet. In praktischer Arbeit analysieren die Studierenden Problemstellungen im Systemumfeld (K1, K2, K4), implementieren Lösungen auf der Grundlage anerkannter Konzepte und Methoden (K3) mit Hilfe von Standardwerkzeugen (K6, K9) und prüfen sie (K7). Sie recherchieren dazu in Dokumentationen (K8, K15) und passen vorhandene Software an (K10).

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. In den Übungen und insbesondere im Praktikum arbeiten die Studierenden in Kleingruppen und verteidigen ihre Lösungen (K8, K13, K16).

Wozu: Systemsoftware, also Betriebssysteme und Dienstsoftware für verteilte Systeme, bietet die Plattform zur Erstellung von Anwendungen, die nebenläufig und verteilt arbeiten - Eigenschaften, die für heutige komplexe Softwaresysteme typisch sind. Entsprechende Programmierkenntnisse und Wissen über die zugehörigen Grundlagen sind somit essentiell für die Erstellung moderner Software (HF1). Durch ihre praktische Programmierarbeit erwerben die Studierenden Erfahrungen, die wichtig sind für die Erfassung von Anforderungen, die Entwicklung von Konzepten zur technischen Lösung und zu ihrer Bewertung (HF2) sowie zur Organisation von Prozessen und zum Betrieb von Systemen, die nebenläufig und verteilt arbeiten (HF3). Die Durchführung im Team mit dem Dozenten als "Auftraggeber" stärkt die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte**Vorlesung / Übungen**

- Grundlagen von Betriebssystemen und Verteilten Systemen
 - Einordnung und Aufgaben eines Betriebssystems im Rechensystem
 - zu verwaltende Betriebsmittel
 - Nebenläufigkeit in Hard- und Software
 - Komponenten und Eigenschaften Verteilter Systeme
 - Software-Strukturen
 - Betriebssystemkern
 - Hierarchische Strukturen
 - Virtuelle Maschinen
 - Client-Server-Systeme
 - Peer-to-Peer-Systeme

Das UNIX/Linux-Betriebssystem - Geschichte und Standards - Schalenstruktur - Kern mit Programmierschnittstelle - Shell mit Benutzerschnittstelle - wichtige Benutzerkommandos - Aufbau des Dateisystems - Programmierung in C

- Nebenläufigkeit
 - Prozesse und Threads
 - grundlegende Eigenschaften
 - Prozesse in UNIX
 - Threads in Java
 - Synchronisation
 - grundlegende Synchronisationsbedingungen
 - wechselseitiger Ausschluss
 - Reihenfolge
 - Mechanismen zur Durchsetzung
 - Interruptsperrung
 - Spinlocks
 - Signale
 - Semaphore
 - Monitore
 - Deadlocks
- Kommunikation
 - Grundbegriffe
 - speicher- vs. nachrichtenbasierte Kommunikation
 - Mailboxen und Ports
 - synchrone vs. asynchrone Kommunikation
 - lokale Kommunikation
 - Shared Memory
 - Message Queues
 - Pipes
 - Kommunikation in verteilten Systemen
 - Protokolle
 - Sockets
- Umgang mit den Schnittstellen eines Betriebssystems
 - zeichenorientierte Benutzerschnittstelle (Konsole)
 - Programmierschnittstelle
- Steuerung nebenläufiger Aktivitäten in einem Betriebssystem
 - von der Benutzerschnittstelle aus
 - durch Funktionen der Programmierschnittstelle
- Synchronisation nebenläufiger Ausführungen durch Synchronisationsmechanismen
- Nutzung verschiedener Kommunikationsmechanismen
 - lokale Mechanismen
 - Mechanismen in Rechnernetzen

Praktikum

- Befehle der zeichenorientierten UNIX/Linux-Benutzerschnittstelle
 - Nutzung durch Eingabe über die Tastatur
 - Nutzung durch Einbettung in Shell Scripts
- insbesondere zur Steuerung nebenläufiger Prozesse
- C-Funktionen der UNIX/Linux-Programmierschnittstelle
 - zum Zugriff auf Dateien und Geräte
 - zur Erzeugung und elementaren Steuerung von Prozessen
 - zur Synchronisation von Prozessen
 - zur Kommunikation von Prozessen (lokal und im Netz) - je nach verfügbarer Zeit
- Anwendung der unter "Kenntnisse (fachliche Inhalte)" genannten Aspekte auf praxisbezogene Szenarien durch selbstständige Arbeit in kleinem Team.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	180 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	135 Stunden
Empfohlene Voraussetzungen	▪ Modul PI1: Sicherer Umgang mit einer Programmiersprache. ▪ Modul PI2: Sicherer Umgang mit einer Programmiersprache. ▪ Modul DR: Kenntnisse über Aufbau und Funktionalität eines Digitalrechners. ▪ Modul GSP: Grundkenntnisse über die hardwarenahe Programmierung eines Digitalrechners. ▪ Modul NP: Grundkenntnisse über Internet-Protokolle. ▪ prozedurale Programmierung ▪ Architektur von Digitalrechnern (Grundkenntnisse) ▪ Internetprotokolle (Grundkenntnisse)
Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: 1 Termin
Empfohlene Literatur	▪ Siehe http://www.nt.fh-koeln.de/vogt/bs/bvs_lit.pdf
Verwendung des Moduls in weiteren Studiengängen	▪ BVS1 in Bachelor Elektrotechnik 2020 ▪ BVS1 in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ BVS1 in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	18.8.2025, 16:18:31

6.7 BVS2 - Verteilte Systeme

Modulkürzel	BVS2_BaTIN2020
Modulbezeichnung	Verteilte Systeme
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	BVS2 - Verteilte Systeme
ECTS credits	5
Sprache	englisch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Pascal Cerfontaine/Professor Fakultät IME
Dozierende*r	Prof. Dr. Pascal Cerfontaine/Professor Fakultät IME

Learning Outcome(s)

Was: Aufbauend auf BVS1 vermittelt das Modul vertiefend Kompetenzen zum Umgang mit Systemsoftware, insbesondere mit Diensten in verteilten Systemen. In praktischer Arbeit analysieren die Studierenden Problemstellungen im Systemumfeld (K1, K2, K4), implementieren Lösungen auf der Grundlage anerkannter Konzepte und Methoden (K3) mit Hilfe von Standardwerkzeugen (K6, K9) und prüfen sie (K7). Sie recherchieren dazu in Dokumentationen (K8, K15) und passen vorhandene Software an (K10).

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. In den Übungen und insbesondere im Praktikum arbeiten die Studierenden in Kleingruppen und verteidigen ihre Lösungen (K8, K13, K16).

Wozu: Systemsoftware, also Betriebssysteme und Dienstsoftware für verteilte Systeme, bietet die Plattform zur Erstellung von Anwendungen, die nebenläufig und verteilt arbeiten - Eigenschaften, die für heutige komplexe Softwaresysteme typisch sind. Entsprechende Programmierkenntnisse und Wissen über die zugehörigen Grundlagen sind somit essentiell für die Erstellung moderner Software (HF1). Durch ihre praktische Programmierarbeit erwerben die Studierenden Erfahrungen, die wichtig sind für die Erfassung von Anforderungen, die Entwicklung von Konzepten zur technischen Lösung und zu ihrer Bewertung (HF2) sowie zur Organisation von Vorgängen und zum Betrieb von Systemen, die nebenläufig und verteilt arbeiten (HF3). Die Durchführung im Team mit dem Dozenten als "Auftraggeber" stärkt die Interaktionsfähigkeit der Studierenden (HF4).

Was: Das Modul vermittelt Wissen über die Implementation von Systemsoftware auf Grundlage einer Hardwarearchitektur. Die Studierenden lernen die Details ihrer Realisierung sowie die zugrundeliegenden Techniken, Konzepte und Strategien kennen. Sie spielen in den Übungen typische Szenarien durch und lernen dabei die Auswirkungen strategischer Entscheidungen bei Entwurf, Implementierung und Ausführung der Systemsoftware kennen (K1, K2, K3, K4, K9).

Womit: Der Dozent vermittelt das grundlegende Wissen in der Vorlesung und leitet in den Übungen zu seiner Anwendung an.

Wozu: Die Systemsoftware ist ein zentraler Bestandteil eines jeden Rechensystems und somit entscheidend für seine Einsatzmöglichkeiten und Leistung. Entsprechendes Wissen über ihre Eigenschaften und mögliche Alternativen bei ihrer Realisierung ist daher essentiell für die Erstellung (HF1), Analyse (HF2) und Organisation (HF3) informationstechnischer Systeme.

Modulinhalte**Vorlesung / Übungen**

Kooperation
Client-Server-Modell
Beispiele: Namens- und Dateidienste
geschichtete Architekturen
Peer-to-Peer-Modell
prozedurale Kooperation: Remote Procedure Call
objektorientierte Kooperation
Remote Method Invocation
objektorientierte Middleware
Web-basierte Dienste
dynamische Web-Seiten
Web Services

Implementierung von Software-Nebenläufigkeit
Verwaltung und Steuerung von Prozessen
Dispatching und Scheduling
Exceptions und Interrupts
Speicherkonzepte
Komponenten der Speicherhierarchie
Swapping
Virtueller Speicher
Prozesse in Verteilten Systemen
Lastverteilung, Fehlertoleranz, Synchronisation

Dateisysteme
logische und reale Strukturen
lokale Dateisysteme
Implementierung von Verzeichnissen
Organisation der Festplatte
Leistungssteigerung und Fehlertoleranz
verteilte Dateisysteme
File Server und Name Server
Verteilte Dateibäume
Caching und Replikation

Beurteilung verschiedener Verfahren und Techniken zum Prozessor-Scheduling, zur Verwaltung von Speicherhierarchien, zur Implementierung lokaler und verteilter Dateisysteme

Programmierung von und mit Diensten in lokalen und verteilten Systemen

Dienste in verteilten Systemen
Grundlagen von Cloud Computing und Web Services
Apache-basierte Systeme
kommerziell verfügbare Systeme

Praktikum

C-Funktionen der UNIX/Linux-Programmierschnittstelle zur Kommunikation und Kooperation lokal und im Internet durch Nutzung von Shared Memory, Message Queues und Sockets
durch Remote Procedure Call

Java-Techniken zur Kommunikation und Kooperation
Web Services: SOAP, REST
ggf. andere (wird kurzfristig festgelegt)

Anwendung der unter "Kenntnisse (fachliche Inhalte)" genannten Aspekte auf praxisbezogene Szenarien durch selbstständige Arbeit in kleinem Team.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
-------------------------------	---

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul PI1: Sicherer Umgang mit einer Programmiersprache. ▪ Modul PI2: Sicherer Umgang mit einer Programmiersprache. ▪ Modul DR: Kenntnisse über Aufbau und Funktionalität eines Digitalrechners. ▪ Modul GSP: Grundkenntnisse über die hardwarenahe Programmierung eines Digitalrechners. ▪ Modul NP: Grundkenntnisse über Internet-Protokolle. ▪ Modul BVS1: Sämtliche Modulinhalte, da BVS2 eine unmittelbare Fortsetzung von BVS1 ist ▪ prozedurale Programmierung - Architektur von Digitalrechnern (Grundkenntnisse) - Internetprotokolle (Grundkenntnisse) - Sämtliche Inhalte von BVS1
Zwingende Voraussetzungen	
Empfohlene Literatur	▪ Siehe http://www.nt.fh-koeln.de/vogt/bs/bvs_lit.pdf
Verwendung des Moduls in weiteren Studiengängen	▪ BVS2 in Bachelor Elektrotechnik 2020 ▪ BVS2 in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ BVS2 in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.8 BWR - Betriebswirtschaft und Recht

Modulkürzel	BWR_BaTIN2020
Modulbezeichnung	Betriebswirtschaft und Recht
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	BWR - Betriebswirtschaft und Recht
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4
Häufigkeit des Angebots	jedes Semester
Modul-Verantwortliche*r	Prof. Dr. Stefan Kreiser/Professor Fakultät IME
Dozierende*r	Dr. Diana Pülpichhuysen/Lehrbeauftragte

Learning Outcome(s)

1. Fachkompetenzen (lernergebnisorientiert)

- Die Studierenden können eine eigene Business Idee generieren, mit Hilfe von Business Modelling entwickeln und validieren.
- Sie kennen die zentralen Inhaltsfelder der BWL und deren Bedeutung für Entre- und Intrapreneure.
- Sie wissen, was notwendig ist, um ein Unternehmen funktionsfähig aufzubauen und Ziel- und zukunftsorientiert zu betreiben.
- Sie kennen die für Unternehmensgründungen relevanten rechtlichen Rahmenbedingungen und können darauf aufbauend passende Entscheidungen treffen.
- Sie sind damit grundsätzlich in der Lage, betriebswirtschaftliche Problemstellungen zu analysieren, Lösungsvorschläge zu entwickeln und (theoretisch) auszuführen.

2. Fachübergreifende Kompetenzen

: Die Studierenden können im Team projektartig vorgegebene Ziele erreichen. Sie wenden hierzu erlerntes, theoretisches Wissen auf ein Praxisbeispiel an (Transferkompetenz). Sie können:

- die notwendige Literatur recherchieren, lesen und verstehen
- mit anderen Menschen zusammenzuarbeiten und gemeinsam Ziele erreichen,
- ein komplexes Arbeitsergebnis vor Publikum präsentieren sowie
- sich selbst reflektieren und Leistungen anderer bewerten.

Die Studierenden verfügen somit über

- methodisches Grundlagenwissen der Disziplinen BWL, Recht und Entrepreneurship,
- Selbst-, Sozial und Reflexionskompetenz,
- Präsentations- und Diskussionsfähigkeit.

Modulinhalte**Projekt**

Anhand einer fiktiven Unternehmensgründung (Business Modelling) erlangen die Studierenden anwendungsbezogen die relevanten Kenntnisse und Fähigkeiten aus den Disziplinen BWL, Recht und Entrepreneurship.

Vorlesung

1. Business Ideation
2. Business Modelling (durchgehend)
3. Marktanalyse, Kundengruppenanalyse, Stakeholderanalyse
4. betriebliche Leitungsprozesse
5. Rechtliche Rahmenbedingungen, Steuern
6. Kostenrechnung, Preiskalkulation
7. Externes Rechnungswesen
8. Business Model Evaluierung (SWOT-Analyse)

Weitere, spezielle Unterrichtseinheiten zu:

1. Selbst- und Teammanagement
2. Präsentationstechnik
3. Experience Report eines Unternehmers/einer Unternehmerin

Lehr- und Lernmethoden	▪ Projekt ▪ Vorlesung
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	34 Stunden $\triangleq$ 3 SWS
--------------------	-------------------------------

Selbststudium	116 Stunden
----------------------	-------------

Empfohlene Voraussetzungen**Zwingende Voraussetzungen**

Empfohlene Literatur	▪ Hölter, E. (2018): Betriebswirtschaft für Studium, Schule und Beruf. Stuttgart: Schäffer-Poeschel. ▪ Osterwalder, A. & Pigneur, Y. (2010): Business Model Generation. Hoboken, New Jersey: John Wiley & Sons.
-----------------------------	--

Verwendung des Moduls in weiteren Studiengängen	▪ BWR in Bachelor Elektrotechnik 2020 ▪ BWR in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ BWR in Bachelor Medientechnologie 2020 ▪ BWR in Bachelor Medientechnologie 2024 ▪ BWR in Bachelor Optometrie 2021 ▪ BWR in Bachelor Informatik und Systems-Engineering 2024
--	--

Perma-Links zur Organisation	<u>llu</u>
-------------------------------------	----------------------------

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.9 CA - Computeranimation

Modulkürzel	CA_BaTIN2020
Modulbezeichnung	Computeranimation
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	CA - Computeranimation
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Stefan Grünvogel/Professor Fakultät IME
Dozierende*r	Prof. Dr. Stefan Grünvogel/Professor Fakultät IME

Learning Outcome(s)

WAS:

Die mathematischen, algorithmischen und theoretischen Grundlagen der Computeranimation erklären können, schriftlich und mündlich, unter Verwendung der entsprechenden Fachtermini.

WOMIT:

Die entsprechenden Grundlagen werden nach dem Prinzip des Flipped Classrooms vermittelt und zunächst in Form von einfachen Aufgaben (ohne Hilfe von Software) schriftlich geübt.

WOZU:

Um Anwendungen und Software zur Computeranimation nicht nur als Black Box zu verwenden, sondern auch deren Arbeitsweise zu verstehen und sich selbstständig in weiterführende (wissenschaftliche) Themengebiete der Computeranimation einarbeiten zu können.

WAS:

Eine Problemstellung oder Aufgabenstellung aus dem Bereich der Computeranimation analysieren und die passenden Methoden und Verfahren auswählen zu können.

WOMIT:

Im Praktikum wird schrittweise an die Herangehensweise zur Lösen von Aufgabenstellungen in der Computeranimation herangeführt und typische Lösungsansätze vermittelt. Dazu notwendige fachlichen Kenntnisse werden per Flipped Classroom vermittelt.

WOZU:

Um Verfahren, Algorithmen und Geräten zur Produktion, Speicherung, Übertragung, Verarbeitung, Wiedergabe und Präsentation von Computeranimation analysieren und bewerten zu können.

WAS:

Methoden und Software der Computeranimation anwenden, weiterentwickeln oder selbst entwickeln.

WOMIT:

Im Praktikum werden schrittweise an Hand einer Game Engine oder einer Softwarebibliotheken die Kenntnisse in Form praktischer Übungsaufgaben vertieft und die Implementierung von Software zur Computeranimation geübt.

WOZU:

Um Verfahren, Algorithmen und Geräte zu Produktion und Wiedergabe von Computeranimation entwickeln und integrieren können.

Modulinhalte***seminaristischer Unterricht***

Animationssysteme

- Hierarchien in Szenen
- Animationssystem
- Zeit und Game Loop

Objektanimation

- Bewegung im Raum
- Steuerung von Zeit, Geschwindigkeit und Wegstrecke
- Interpolation
- Rotationen

Characteranimation

- Kinematik
- Skinning
- Blend Shapes
- Motion Capture
- Bearbeitung von Bewegungsdaten

Prozedurale Animation

- Physikalisch basierte Animation
- Partikelsysteme

Lehr- und Lernmethoden	seminaristischer Unterricht
-------------------------------	-----------------------------

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	23 Stunden $\pm$ 2 SWS
--------------------	------------------------

Selbststudium	127 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul MA1: Problemlösungskompetenz aus dem Bereich Analysis einer Veränderlichen. Sicheres Beherrschen der entsprechenden Symbole und Formalismen ▪ Modul MA2: Problemlösungskompetenz aus dem Bereich lineare Algebra und Analysis mehrerer Veränderlichen sowie Differentialgleichungen. Sicheres Beherrschen der entsprechenden Symbole und Formalismen. ▪ Modul PI1: Entwickeln von Programmen zur Lösung konkreter Problemstellungen, abstrahieren von Problembeschreibungen in Algorithmen und überprüfen von Programmen auf Fehler. ▪ Modul PI2: Entwerfen und verwenden objekt-orientierter Modelle und dynamischer Datenstrukturen zu einer gegebenen Problemstellung und Umsetzung in einer Programmiersprache. Lösen von Problemstellung mittels geeigneter Algorithmen ▪ Grundkenntnisse Computergrafik Programmierkenntnisse vermittelt im Umfang der Informatik 1 und Informatik 2 Sicherer Umgang mit lineare Algebra sowie Analysis einer und mehrer Veränderlicher um Umfang der Kenntnisse aus Mathematik 1 und Mathematik 2
-----------------------------------	--

Zwingende Voraussetzungen

Empfohlene Literatur	▪ Stefan M. Grünvogel, Einführung in die Computeranimation, Springer, 2024 ▪ Rick Parent, Computer Animation: Algorithms and Techniques, Morgan Kaufmann, 2007, ▪ Dietmar Jackel et. al., Methoden der Computeranimation, Springer, 2006 ▪ Jason Gregory, Game Engine Architecture, AK Peters, 2009
-----------------------------	--

Verwendung des Moduls in weiteren Studiengängen	▪ CA in Bachelor Medientechnologie 2020 ▪ CA in Bachelor Medientechnologie 2024 ▪ CA in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise

Letzte Aktualisierung 19.7.2025, 14:32:16

6.10 CG - Computergrafik

Modulkürzel	CG_BaTIN2020
Modulbezeichnung	Computergrafik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	CG - Computergrafik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr.-Ing. Arnulph Fuhrmann/Professor Fakultät IME
Dozierende*r	Prof. Dr.-Ing. Arnulph Fuhrmann/Professor Fakultät IME

Learning Outcome(s)

Das Modul vermittelt folgende Kenntnisse und Fertigkeiten:

- Beschreiben von Methoden zum geometrischen Modellieren
- Erklären von Transformationen
- Beschreiben der grundlegenden Graphikhardware
- Beschreiben der einzelnen Stufen der Rendering Pipeline
- Erklären von globalen und lokalen Beleuchtungsmodellen
- Beschreiben von Methoden zur Texturierung
- Gegenüberstellen der behandelten Beleuchtungsmodelle
- Entscheiden welches Verfahren geeignet ist, um eine konkrete Problemstellung der Computergrafik zu lösen
- Entwickeln von Computergrafikanwendungen (Verwenden eines 3D-APIs, Erstellen interaktiver 3D-Programme, Anwenden der mathematischen Basis der Computergrafik, Anwenden der grundlegenden Algorithmen der Computergrafik, Testen und Debuggen von Anwendungen)

Die Kompetenzen werden zunächst über eine Vorlesung durch den Dozenten vermittelt und danach im Praktikum von den Studierenden vertieft.

Die sichere Anwendung der Grundlagen der Computergrafik ist Voraussetzung für die Entwicklung interaktiver medientechnischer Systeme (HF1, HF2) und erlaubt die Bewertung bestehender Systeme (HF2).

Modulinhalte**Vorlesung / Übungen**

Geometrisches Modellieren
 Transformationen
 Graphikhardware
 Rendering Pipeline
 Lokale Beleuchtungsmodelle
 Texturen
 Globale Beleuchtungsmodelle
 Volume Rendering
 Shaderprogrammierung
 Fortgeschrittene Texturierungstechniken
 Tonemapping

Praktikum

- Entwickeln von Computergrafikanwendungen
- Erstellen interaktiver 3D-Programme
- Verwenden eines 3D-APIs
- Anwenden der mathematischen Basis der Computergrafik
- Anwenden der grundlegenden Algorithmen der Computergrafik
- Testen und debuggen der eigenen Anwendung
- Textuelle Aufgabenstellungen erfassen und verstehen

Lehr- und Lernmethoden

- Vorlesung / Übungen
- Praktikum

Prüfungsformen mit Gewichtung siehe Prüfungsordnung

Workload 150 Stunden

Präsenzzeit 45 Stunden $\pm$ 4 SWS

Selbststudium 105 Stunden

Empfohlene Voraussetzungen Programmierkenntnisse
Mathematik 1 und 2

Zwingende Voraussetzungen Praktikum erfordert Anwesenheit im Umfang von: 3 Termine

Empfohlene Literatur

- P. Shirley, S. Marschner: Fundamentals of Computer Graphics, Fifth Edition, AK Peters, 2021
- T. Akenine-Möller, et al.: Real-Time Rendering, Taylor & Francis Ltd., 2018
- M. Pharr, W. Jakob, and G. Humphreys, Physically Based Rendering: From Theory To Implementation, Morgan Kaufmann, 4. Edition, 2023

Verwendung des Moduls in weiteren Studiengängen

- CG in Bachelor Medientechnologie 2020
- CG in Bachelor Medientechnologie 2024
- CG in Bachelor Informatik und Systems-Engineering 2024

Besonderheiten und Hinweise

Letzte Aktualisierung 19.7.2025, 14:32:16

6.11 DB1 - Datenbanken 1

Modulkürzel	DB1_BaTIN2020
Modulbezeichnung	Datenbanken 1
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	DB1 - Datenbanken 1
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	3
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Andreas Behrend/Professor Fakultät IME
Dozierende*r	Prof. Dr. Andreas Behrend/Professor Fakultät IME

Learning Outcome(s)

WAS? HF1: Studierende sollen den Aufbau von Datenbanksystemen zur Speicherung und Verarbeitung von Informationen kennenlernen. Sie sollen relationale Datenbanken erstellen und Anfragen mittels SQL auf diese Datenbanken programmieren können. Dabei sollen Sie auch in der Lage sein, diese Anfragen in andere Programmiersprachen einzubetten (z.B. SQL-Anfragen in Java mittels der JDBC-Schnittstelle einbetten). Sie sollen in der Lage sein, den Datenaustausch mit benachbarten Softwaresystemen über definierte Austauschformate (z.B. XML) realisieren zu können.

HF2: Gegebene Anforderungskataloge für zu entwickelnde Softwaresysteme sollen auf ihren Bedarf an persistenten Daten analysiert werden können. Dabei sollen unterschiedliche Persistenzmechanismen analysiert werden können. Hierzu sollen verschiedene Datenbankmodelle im Überblick kennengelernt werden. In Bezug auf relationale Datenbanken als Zielsysteme sollen ERD-Modelle entwickelt und normalisiert werden können.

HF3: Kleinere Datenbanksysteme, die nach analytischen Vorgaben selbst entwickelt wurden, sollen mit Schnittstellen zu Nachbarsystemen organisiert und betrieben werden können.

WOMIT? Vortrag zu HF1, HF2 und HF3. Üben an Hand praktischer Beispiele zu HF1, HF2 und HF3. Drei kleinere Projekte in Laborversuchen zu HF1 und HF3, dabei sollen die Voraussetzungen zur Erstellung der Lösungen für HF1 und HF3 mittels Analysetechniken aus HF2 spezifiziert und anschließend dokumentiert werden können.

WOZU? In Softwareabteilungen großer Industrie- und Dienstleistungsunternehmen und bei Unternehmensberatungen für Soft- und Hardwaresystemen spielt die Entwicklung von Datenbanksystemen eine sehr große Rolle. Hier werden Informatiker dringend benötigt, die Datenbanken entwerfen, hierauf bezogene Anfrageprogramme entwickeln und testen können und Datenbanken in Betrieb halten können.

Modulinhalte

Vorlesung / Übungen

Allgemeines Datenbankmodell
 Relationales Datenbanksystem
 SQL
 Einbettung von SQL in eine höhere Programmiersprache (z. B. JDBC)
 Datenbankspezifikation und Design
 ERD
 Normalformen
 XML und DB
 DTD

Praktikum

Entwicklung einer einfachen Datenbank mit mehreren Tabellen auf Grundlage eines Anforderungskatalogs; Modellierung von Abhängigkeiten mit Fremdschlüsselbeziehungen; Programmierung komplexer DB-Anfragen mit JDBC-Programmen; Spezifikation einer Datenbank mit ERD; Definition von DB/XML-Schnittstellen mit einer DTD.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\triangleq$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul MA1: Mengen, kartesisches Produkt, Relationen ▪ Grundstudium Informatik ▪ Grundstudium Mathematik
Zwingende Voraussetzungen	
Empfohlene Literatur	▪ G. Vossen: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme ▪ A. Kemper, A. Eickler: Datenbanksysteme ▪ C. Türker: SQL 1999 & SQL 2003
Verwendung des Moduls in weiteren Studiengängen	▪ DB in Bachelor Elektrotechnik 2020 ▪ DB1 in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ DB1 in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.12 DB2 - Datenbanken 2

Modulkürzel	DB2_BaTIN2020
Modulbezeichnung	Datenbanken 2
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	DB2 - Datenbanken 2
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Andreas Behrend/Professor Fakultät IME
Dozierende*r	Prof. Dr. Andreas Behrend/Professor Fakultät IME

Learning Outcome(s)

WAS? HF1: Studierende sollen neben dem relationalen auch andere Datenbanksysteme kennenlernen: objektorientierte, objektrelationale und NoSQL-Datenbanksysteme. Weiterhin sollen sie neben dem ERM weitere Datenbank-Design Methoden kennenlernen, z.B. abstrakte Datentypen (ADT). Die Studierenden sollen die wichtigsten Arten von NoSQL-Datenbanksystemen kennenlernen: Key-Value-Datenbanken, Wide Column Stores, dokumentorientierte Datenbanken. Sie sollen die unterschiedlichen Schema Anforderungen analysieren lernen sowie die unterschiedlichen nicht-relationalen Datenmodelle, wie z.B. der JSON-Dokumenttyp bei dem NoSQL-Datenbanksystem CouchDB. Die Studierenden sollen objektrelationale Datenbanken und NoSQL Datenbanken erstellen und Anfragen auf diese Datenbanken programmieren können. Dabei sollen Sie auch in der Lage sein, Schnittstellen auf der Basis unterschiedlicher Austauschformate (z.B. JSON und XML) unter Berücksichtigung unterschiedlicher Grammatikmodelle (bei JSON: JSON-Schema, bei XML: XML-Schema) programmieren zu können. Weiterhin sollen Sie Bayer-Bäume als wesentliche Datenstruktur für die Sekundärspeicherverwaltung kennenlernen

HF2: Gegebene Anforderungskataloge für zu entwickelnde Datenstrukturen sollen sowohl objektrelationale als auch in Hinblick auf dokumentenorientierte NoSQL Datenbanken modelliert werden können. Hierbei soll das Konzept abstrakter Datentypen angewendet werden können. Das Leistungsverhalten von Algorithmen der Sekundärspeicherverwaltung, die auf Bayer-Bäumen basieren, sollen im Unterschied zu anderen Strukturen der Sekundärspeicherverwaltung (z.B. ISAM) bewertet werden können. Unterschiedliche Grammatikmodelle für Austauschformate sollen zum validierenden Parsen von XML- oder JSON-Daten für Datenbankschnittstellen angewendet werden können.

HF3: Objektrelationale und NoSQL Datenbanksysteme, die nach analytischen Vorgaben selbst entwickelt wurden, sollen mit Schnittstellen zu Nachbarsystemen organisiert und betrieben werden können.

WOMIT? Vortrag zu HF1, HF2 und HF3. Üben an Hand praktischer Beispiele zu HF1, HF2 und HF3. Drei kleinere Projekte in Laborversuchen zu HF1, HF2 und HF3, dabei sollen die Voraussetzungen zur Erstellung der Lösungen für HF1, HF2 und HF3 mittels der in HF2 genannten Grammatikmodelle und dem in HF1 genannten Konzept abstrakter Datentypen HF2 spezifiziert und nach Implementation dokumentiert werden können.

WOZU? In großen Unternehmen und auf größeren Internetplattformen (z.B. Google, Ebay, Amazon) spielen skalierbare, hochperformante und föderierte Cloud-Datenbanken eine große Rolle. Diese Datenbanken operieren mit NoSQL- und ADT-Konzepten. Industrie- und Dienstleistungsunternehmen, die bei der Umsetzung von Industrie 4.0 Strategien Cloud-Datenbanken anwenden oder entwickeln möchten, benötigen dringend Informatiker, die z.B. NoSQL Datenbanken entwerfen, hierauf bezogene Anfrageprogramme entwickeln, testen und in Betrieb halten können.

Modulinhalte**Vorlesung / Übungen**

XML Grammatiken mit XML Schema aufstellen
Abstrakte Datentypen
Objektorientierte Datenbanken
Objektrelationale Datenbanken
NoSQL Datenbanken
Bayer Bäume

Praktikum

Aufstellen von XML Schemata, von abstrakten Datentypen, von objekt-relationalen Datentypen. Programmierung von Anfragen an objektrelationale und NoSQL Datenbanken. Validierendes Parsen von JSON Dokumenten.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
-------------------------------	---

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\triangleq$ 4 SWS
--------------------	-------------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul DB1: Relationale Datenbanken, SQL, XML mit DTD, JDBC. ▪ Modul FSA: Klassen formaler Grammatiken nach Chomsky. ▪ Modul BVS1: Funktionalität und Aufbau des Dateisystems als Teil des Betriebssystemkerns. ▪ Modul SE: Prinzipien und Methoden der Spezifikation verteilter Softwaresysteme. ▪ Grundstudium Informatik ▪ Grundstudium Mathematik ▪ Datenbanken 1
-----------------------------------	--

Zwingende Voraussetzungen

Empfohlene Literatur	▪ C. Türker: SQL 1999 & SQL 2003 ▪ St. Edlich: NoSQL Datenbanken
-----------------------------	--

Verwendung des Moduls in weiteren Studiengängen	DB2 in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.13 DM - Data Mining

Modulkürzel	DM_BaTIN2020
Modulbezeichnung	Data Mining
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	DM - Data Mining
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Beate Rhein/Professor Fakultät IME
Dozierende*r	Prof. Dr. Beate Rhein/Professor Fakultät IME

Learning Outcome(s)

Was:

Methoden des maschinellen Lernens auf typische Datensätze der technischen Informatik anwenden
gängige Fallstricke des Data Mining in der Vorgehensweise kennen
für eine Aufgabenstellung das geeignete Verfahren bestimmen können
Qualität von Datensätzen beurteilen
Datenschutzgesetze kennen
weit verbreitete Software hierfür anwenden
eigenverantwortliches Arbeiten lernen

Womit:

Die Methoden werden anhand eines Vortrags oder per Lernvideos vermittelt und in Vorlesung und Übung direkt angewendet. Jeder Student wird ein kleines Projekt durchführen (je nach Anzahl der Studierenden in Gruppenarbeit).

Wozu:

Data Mining wird bei den späteren Arbeitgebern immer mehr eingeführt, etwa in der Robotik, aber auch zur Überwachung und Steuerung von Produktionsprozessen oder Energiesystemen und zur Auswertung von Kundendaten, hier ist ein verantwortlicher Einsatz von Daten wichtig

Modulinhalte**Vorlesung / Übungen**

Einführung in eine geeignete Software, z.B. Python
 Einführung in deskriptive Statistik und evtl. auch Wahrscheinlichkeitsrechnung
 Überwachtes Lernen:
 - Klassifikationsverfahren: Ablauf, Performanzmaße, Anwendung eines Verfahrens des instanzbasierten Lernen, z.B. k-nearest-neighbor und eines Verfahrens des modellbasierten Lernen, z.B. Entscheidungsbäume
 - evtl. Regressionsanalyse: über maschinelles Lernen und klassisch
 Unüberwachtes Lernen:
 - Clusteranalyse: k-means, evtl. auch DBSCAN
 Preprocessing der Daten:
 - Behandlung von beschädigten / fehlenden Daten
 - Ausreißer oder Noise - Problematik
 - Skalierung
 - Visualisierung der Daten
 - evtl. Dimensionsreduzierung
 - Beurteilung der Qualität der Daten
 - evtl. verschiedene Arten von Datensätzen betrachten, Bezug zu NoSql-Datenbanken herstellen

Ausblick auf aktuelle Forschung, z.B. Bilderkennung, Natural Language Processing, Reinforcement Learning

zu Aufgabenstellungen geeignete Methode und Gesamtvorgehensweise nennen und anwenden können
 geeignetes Performanzmaß wählen und beurteilen
 Datenschutzrichtlinien anwenden

Übungen / Praktikum

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Übungen / Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	57 Stunden $\pm$ 5 SWS
Selbststudium	93 Stunden
Empfohlene Voraussetzungen	▪ Modul MA1: mathematische Modelle verstehen und aufstellen Differentialrechnung ▪ Modul MA2: Funktionen mit mehreren Veränderlichen anwenden Lineare Algebra: Matrizen aufstellen und mit ihnen rechnen ▪ Modul PI1: Grundlagen der Programmierung beherrschen ▪ Aus Mathematik 1 und 2 wird die Fähigkeit benötigt, mathematische Modelle aufzustellen, sowie Kenntnisse der Differentialrechnung und der Linearen Algebra.
Zwingende Voraussetzungen	Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Übungen / Praktikum
Empfohlene Literatur	▪ A. Geron: Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme, Heidelberg, o'Reilly Verlag 2017, 978-3960090618 ▪ S. Raschka, V. Mirjalili: Machine Learning mit Python und Scikit-Learn und TensorFlow: Das umfassende Praxis-Handbuch für Data Science, Predictive Analytics und Deep Learning, mitp Verlag, 2018, 978-3958457331 ▪ J. Frochte, Jörg: Maschinelles Lernen, München, Carl Hanser Verlag GmbH & Co. KG, 2018, eBook ISBN: 978-3-446-45705-8, Print ISBN: 978-3-446-45291-6 ▪ A. Müller: Einführung in Machine Learning mit Python: Praxiswissen Data Science, Heidelberg, o'Reilly Verlag 2017, eBook: 978-3-96010-111-6

Verwendung des Moduls in weiteren Studiengängen	▪ DML in Bachelor Elektrotechnik 2020 ▪ DM in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ DM in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise

Letzte Aktualisierung 19.7.2025, 14:32:16

6.14 DR - Digitalrechner

Modulkürzel	DR_BaTIN2020
Modulbezeichnung	Digitalrechner
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	DR - Digitalrechner
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	1
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Tobias Krawutschke/Professor Fakultät IME
Dozierende*r	Prof. Dr. Tobias Krawutschke/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt die grundlegenden Methoden und Systeme der Digitaltechnik sowie den Entwurf digitaltechnischer Systeme unter Verwendung programmierbarer Bausteinen. Dies geschieht insbesondere auch mit dem Ziel, dass die Studierenden das Prinzip, den Aufbau und die Funktionsweise eines Digitalrechners verstehen und in Form eines rudimentären Von-Neumann-Rechners auch selbst entwickeln und mittels Maschinensprache programmieren können. Aufbauend und vergleichend zu den rudimentären Von-Neumann-Rechner erlernen die Studierenden die grundlegende prinzipielle Funktionsweise einer gängigen CPU (z.B. IA32E-Architektur). Die Studierenden werden in die Lage versetzt, fachspezifische Begriffe, Tools und Techniken im praktischen Umfeld sicher anzuwenden. Aufbauend auf den in der Vorlesung vermittelten Kenntnissen werden komplexere Problemstellungen analysiert, auf Teilsysteme heruntergebrochen und modelliert. Darauf aufbauend wird die Problemlösung mittels Entwurfswerkzeugen implementiert, simuliert, getestet und am Zielsystem in Betrieb genommen.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. Im Praktikum erarbeiten die Studierenden in Kleingruppen Problemlösungen und verteidigen diese.

Wozu: Kompetenzen in der Entwicklung digitaltechnischer Systeme und hier insbesondere auch von Digitalrechnern sind essentiell für technische Informatiker, die im HF 1 arbeiten wollen. Durch die Entwicklung von Problemlösungen erwerben die Studierenden zudem Erfahrungen, die essentiell für das HF 2 sind. Eine projektorientierte Durchführung der Praktika in kleinen Teams mit dem Dozenten als "Auftraggeber" initiiert die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte**Vorlesung / Übungen**

- Boolesche Algebra, Schaltplan/Schaltnetze, Gleichung, Funktionstabelle, typische Schaltnetze wie Multiplexer, Addierer
- Zahlensysteme, Dezimal, Dual, Hexadezimal, Oktal
- Zahlendarstellung in Rechnersystemen, Zweierkomplement, Fixkomma- und Gleitkommadarstellung
- Speicherelemente, taktzustandsgesteuert, taktflankengesteuert, Register, setup-, hold- time
- synchrone Zähler und deren Spezifikation in VHDL
- synchrone Schaltwerke (Automaten) und deren Spezifikation in VHDL
- Zustandsüberführungsdiagramme, Moore, Determinismus, Vollständigkeit
- VHDL, Syntax, Strukturelemente, Datentypen, Vektoren, Verhaltens- und Hierarchische Beschreibung
- Programmierbare Bausteine (FPGA), Simulation, Synthese
- Grundaufbau und Arbeitsweise eines einfachen Rechnersystems, ALU, Steuerwerk, Speicher, I/O
- Programmabarbeitung, Registertransfers, Assemblerprogrammierung
- Aufbau und Funktionsweise einer dedizierten CPU, Architektur, Stack

Projekt

- Analyse von digitalen Schaltungen, Schaltnetzen und Funktionstabellen, Übersetzung in Hardwarebeschreibungssprache, Simulation, Synthese und Place and Route mit Werkzeugen für das Hardware-Design mit realer Zieltechnologie in Form von FPGA-Boards
- Analyse und Implementation von Schaltwerken anhand von Zählern, Teststimuli Entwicklung und Simulation mit anschließender Umsetzung auf FPGA-Boards
- Analyse von Automaten und Umsetzung und Simulation in einer Hardwarebeschreibungssprache als Schaltwerk
- Simulation einer Lehr-CPU und debugging anhand der Simulationsergebnisse und Inbetriebnahme der Lehr-CPU auf FPGA-Boards, sowie Entwicklung von Assemblercode für das reale System.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen
	▪ Projekt

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	keine
-----------------------------------	-------

Zwingende Voraussetzungen	Projekt erfordert Anwesenheit im Umfang von: 5 Termine
----------------------------------	--

Empfohlene Literatur	▪ Urbanski K., Woitowicz R.: Digitaltechnik, 4. Auflage Springer 2004
	▪ Beuth K.: Elektronik Bd. 4 Digitaltechnik, Vogel Verlag 2001
	▪ Lipp H.M.: Grundlagen der Digitaltechnik, 4. Auflage Oldenbourg 2002
	▪ Tanenbaum A. S.; Austin T.: Rechnerarchitektur: Von der digitalen Logik zum Parallelrechner Pearson Deutschland 2014

Verwendung des Moduls in weiteren Studiengängen	DR in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.15 EG - Elektrotechnische Grundlagen für die Technische Informatik

Modulkürzel	EG_BaTIN2020
Modulbezeichnung	Elektrotechnische Grundlagen für die Technische Informatik
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	EG - Elektrotechnische Grundlagen
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	1
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Kai Kreisköther/Professor Fakultät IME
Dozierende*r	▪ Prof. Dr. Kai Kreisköther/Professor Fakultät IME ▪ Prof. Dr.-Ing. Dirk Poggemann/Professor Fakultät IME

Learning Outcome(s)

- **Was:** Die Studierenden analysieren die wesentliche Funktionsweise von elektrotechnischen und elektronischen Systemen unter Einwirkung von zeitunveränderlichen und zeitveränderlichen Spannungen und Strömen. Sie sind in der Lage, deren Verhalten einzuordnen und abzuschätzen. Dies umfasst grundlegendes Wissen über Spannung, Strom, Widerstand, Quellen, Kirchhoffsche Gesetze, Wechselstromkreise, passive und aktive Bauelemente (Diode, Transistor, Operationsverstärker), Hoch- und Tiefpässe, Schwingkreise, Transformatoren, Messtechnik elektrischer Größen, Digitaltechnik, A/D- und D/A-Wandlung, Halbleiterspeicher sowie Signalübertragung auf Leitungen. Zudem sind sie befähigt, sich in diesem Themengebiet fachlich auszutauschen.
- **Womit:** Das notwendige Wissen und grundlegende Fertigkeiten werden durch Vorlesungen und Übungen vermittelt. In betreuten Praktikumsversuchen vertiefen und wenden die Studierenden ihre Kenntnisse praktisch an.
- **Wozu:** Elektrotechnische Systeme bilden die technologische Basis für viele Informationssysteme und sind in zahlreichen technischen Anwendungsbereichen relevant. Ein grundlegendes Verständnis dieser Systeme ist für Studierende technischer Fächer unerlässlich, um moderne Technologien zu verstehen und in interdisziplinären Kontexten effektiv zu kommunizieren, auch wenn die Entwicklung solcher Systeme nicht im Fokus ihrer Tätigkeit steht.

Modulinhalte**Vorlesung / Übungen**

Die Studierenden können elektrotechnische und auch elektronische Systeme mit zeitunveränderlichen und auch zeitveränderlichen Spannungen und Strömen hinsichtlich der wesentlichen Funktionsweise beschreiben, analysieren und deren Verhalten einordnen und abschätzen. Insbesondere sind Studierenden in der Lage, Analysen zu folgenden Themen durchzuführen:

- Widerstand
- Netzwerkanalyse
- Spannungs- und Stromquellen
- Die Kirchhoffschen Sätze, Reihen- und Parallelschaltung
- Leistung und Wirkungsgrad
- Reale Quellen inkl. Arbeitspunkt
- Wechselspannung und -strom
- Spulen und Kondensatoren (komplexe Größen und deren Berechnung)
- Scheinleistung und Blindleistung
- Signal-Übertragung auf Leitungen
- Schaltvorgänge in einfachen RC-Netzwerken
- Transformator
- Dioden (inkl. Photo-, Z-Dioden) und Transistoren
- Operationsverstärker und entsprechende grundlegende Schaltungen
- Digitaltechnik
- A/D- und D/A-Wandlung

Praktikum

Die Studierenden führen projektähnlich elektrotechnische Versuche im Labor durch, die in einem Zusammenhang stehen. Ziel der vorgegebenen Versuche ist das Verständnis der Funktion und die Vermessung eines elektrotechnischen und/oder elektronischen Systems.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen
	▪ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	keine
-----------------------------------	-------

Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 4 Termine
	▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum

Empfohlene Literatur	▪ Gert Hagman, Grundlagen der Elektrotechnik, AULA-Verlag, ISBN 978-3-89104-747-7
-----------------------------	---

Verwendung des Moduls in weiteren Studiengängen	▪ EG in Bachelor Medientechnologie 2024 ▪ EG in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.16 EKS - Entwicklung komplexer Software-Systeme

Modulkürzel	EKS_BaTIN2020
Modulbezeichnung	Entwicklung komplexer Software-Systeme
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	EKS - Entwicklung komplexer SW-Systeme
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Hans Nissen/Professor Fakultät IME
Dozierende*r	Prof. Dr. Hans Nissen/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt unterschiedliche Kompetenzen, die für die Entwicklung und Pflege komplexer Software-Systeme erforderlich sind: die Anwendung und Beurteilung von Entwurfsmustern (K.1, K.3 K.4, K.5, K.9, K.10), die Anwendung von Ansätzen zur professionellen Code-Entwicklung (K.6, K.9), der Einsatz und die Beurteilung von Verfahren zur statischen Code-Analyse (K.4, K.7, K.9), die Beherrschung fortgeschrittene Java-Konzepte (K.5, K.6), der Entwurf und die Realisierung modularisierter Software-Architekturen (K.1, K.3, K.5, K.10), die Einordnung und der Einsatz komplexer Testverfahren (K.2, K.7, K.9), die Fähigkeit, komplexe fachbezogene Probleme zu sehen und Lösungen gegenüber Fachleuten mündlich argumentativ zu vertreten und mit anderen Studierenden weiterzuentwickeln (K.3, K.5, K.7, K.9, K.11, K.16).

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs/Übungsteil unter Verwendung einer Fallstudie und verschiedenen praktischen Demonstrationen.

Im zugehörigen Praktikum erarbeiten die Studierenden in Kleingruppen Lösungen zu vorgegebenen Problemen und wenden dabei die Themenbereiche der Vorlesung praktisch an. Hierzu wird eine selbständige Vertiefung einzelner Themenbereiche, insbesondere die Verwendung typischer Werkzeuge, verlangt (K.8, K.9, K.15). An den Präsenzterminen müssen die Studierenden ihre Lösungen erläutern und verteidigen (K.16).

Wozu: Bei der Entwicklung und Pflege moderner Software muss man sich mit einer stetig zunehmenden System-Komplexität auseinandersetzen. Einige typische Gründe für eine hohe Komplexität sind: die Systeme sind sehr umfangreich, es bestehen sehr viele Schnittstellen zu anderen Systemen, es werden viele und zum Teil sehr umfangreiche Frameworks eingesetzt. Für eine erfolgreiche Tätigkeit in diesem Umfeld ist die sichere Beherrschung hierauf ausgerichteter Methoden, Konzepte und Technologien unbedingt erforderlich (HF.1). Die Erweiterung eines komplexen Systems erfordert die umfangreiche Analyse der bestehenden Abhängigkeiten und die Beurteilung der unterschiedlichen Entwurfsalternativen bezüglich ihrer Auswirkungen auf das Gesamtsystem (HF.2). Ein wesentliches Hilfsmittel zur Beherrschung der System-Komplexität stellt die Wahl einer geeigneten Organisationsform der Systems dar (HF.3). Hierfür sind umfangreiche theoretische und praktische Kenntnisse von Organisationsformen im Kleinen (z.B. Entwurfsmuster) und im Großen (z.B. Modularisierung) erforderlich.

Modulinhalte**Vorlesung / Übungen**

Entwurfsmuster
 Prinzipien und Techniken für schwach-gekoppelte Software
 professionelle Code-Entwicklung
 fortgeschrittene Java-Konzepte
 Modul-orientierte Architekturprinzipien
 komplexere Prüfverfahren: statischer Test, dynamischer Test, formale Verifikation
 Entwurfsmuster anwenden und beurteilen
 Ansätze zur professionellen Code-Entwicklung anwenden und beurteilen
 Verfahren zur automatisierten Code-Analyse anwenden und die Ergebnisse interpretieren
 modularisierte Architekturen entwerfen und realisieren
 komplexe Testverfahren einsetzen

Praktikum

Entwurfsmuster in Programmcode umsetzen
 modularisierte Architekturen für umfangreiche Anwendungen erstellen
 automatisierten Code-Review und statische Code-Analyse anwenden
 Testverfahren auswählen und auf Programme anwenden

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul SE: Spezifikation und Modellierung von Systemen und Software mit UML, Modularisierung in Java, einfache Entwurfsmuster, grundlegende Verfahren zum Prüfen von Software, verschiedene Architekturen von Systemen und Software, Grundbegriffe der Qualitätssicherung, Kenntnisse in Versionsverwaltung ▪ Modul PI1: sehr gute praktische und theoretische Kenntnisse der Programmiersprache Java ▪ Modul PI2: sehr gute praktische und theoretische Kenntnisse der Programmiersprache Java ▪ Modul PP: sehr gute praktische und theoretische Kenntnisse der Programmiersprache Java ▪ Spezifikation und Modellierung von Systemen und Software mit UML, Modularisierung in Java, einfache Entwurfsmuster, grundlegende Verfahren zum Prüfen von Software, verschiedene Architekturen von Systemen und Software, Grundbegriffe der Qualitätssicherung, Kenntnisse in Versionsverwaltung, sehr gute praktische und theoretische Kenntnisse der Programmiersprache Java
-----------------------------------	--

Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 3 Termine ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
----------------------------------	---

Empfohlene Literatur	▪ E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns, MITP Verlags GmbH & Co. KG, 2015. ▪ R. C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008. ▪ S. McConnell: Code Complete, Microsoft Press, 2. Auflage, 2004. ▪ M. Fowler: Refactoring: Improving the Design of Existing Code. Addison-Wesley Verlag, 2. Auflage, 2018. ▪ A. Spillner, T. Linz: Basiswissen Softwaretest, dpunkt Verlag, 5. Auflage, 2012 ▪ P. Liggesmeyer: Software-Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum Akademischer Verlag, 2. Auflage, 2009. ▪ H.M. Sneed, M. Winter: Testen objektorientierter Software, Hanser Verlag, 2001. ▪ G. E. Thaller: Software-Metriken einsetzen–bewerten–messen, Verlag Technik, 2000. ▪ H. Sneed, R. Seidl, Manfred Baumgartner: Software in Zahlen - Die Vermessung von Applikationen. Carl Hanser Verlag, 2010. ▪ Standard ISO/IEC 5055, Software Quality Measurement, 2021. ▪ J. Laski, W. Stanley: Software Verification and Analysis, Springer 2019. ▪ J. Goll: Entwurfsprinzipien und Konstruktionskonzepte der Softwaretechnik, Springer Vieweg, 2018. ▪ J. Goll: Architektur- und Entwurfsmuster in der Softwaretechnik, Springer 2014. ▪ K. Beck: Tidy First? Mini-Refactorings für besseres software-Design, O'Reilly, 2024. ▪ J. Ousterhout: Prinzipien des Software-designs, O'Reilly, 2022.
Verwendung des Moduls in weiteren Studiengängen	EKS in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.17 EL - Elektronik

Modulkürzel	EL_BaTIN2020
Modulbezeichnung	Elektronik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	EL - Elektronik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr.-Ing. Christoph Pörschmann/Professor Fakultät IME
Dozierende*r	Prof. Dr. Alexander Utz/Professor Fakultät IME
Learning Outcome(s)	
Grundlegende Schaltungen mit passiven Bauelementen und Halbleitern (Dioden, BIP-Transistoren) kennenlernen und mit Hilfe erlernter Methoden das Verhalten der Schaltung analysieren. Zur Vermittlung dieser Kompetenzen werden Übungen an Hand exemplarischer Schaltungen durchgeführt und praktische Erfahrungen in Laborversuchen gesammelt. Dadurch wird den Studierenden ermöglicht, in der späteren beruflichen Tätigkeit schaltungstechnische Aufgabenstellungen zu analysieren, Lösungskonzepte zu erarbeiten und letztlich schaltungstechnische Systeme zu entwickeln.	
Modulinhalte	
Vorlesung / Übungen	
lineare passive Schaltungen kennen und analysieren Frequenzverhalten rechnerisch bestimmen Verhalten graphisch im Bodediagramm darstellen Schaltungen mit Halbleiterbauelementen (Diode, Transistor) und Operationsverstärkern kennen und dimensionieren	
Praktikum	
Technische Anleitungen lesen und verstehen Schaltungsaufbauten anschließen und betreiben komplexe Aufgaben in beschränkter Zeit bewältigen theoretisches Wissen in reale Schaltungen umsetzen Ergebnisse kritisch beurteilen und bewerten Typische Messsysteme bedienen technische Grundlagen und Zusammenhänge erklären	
Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden

Empfohlene Voraussetzungen	▪ Modul EG: Kennen und Verstehen elektrischer Bauelemente R,L,C und einfacher passiver Schaltungen. Analysieren von Schaltungen durch Knoten- und Maschengleichungen, Technische Zusammenhänge darstellen und erläutern, Kennlinien von Bauelementen verstehen und interpretieren ▪ Modul MA1: Mathematisches Grundwissen sicher anwenden, insbesondere lineare Gleichungen, Termumformungen, Kürzen von Termen, Bruchrechnung ▪ Modul MA2: Lösen einfacher Differential- oder Integralgleichungen ▪ Grundlegende Kenntnisse von passiven Bauelementen, Widerstand, Kondensator, Induktivität Lösung linearer Gleichungen, rechnen mit komplexen Größen
Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 3 Termine ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	▪ "Halbleiter-Schaltungstechnik", U. Tietze et al., ISBN: 978-3-662-48553-8 ▪ "Design of Analog CMOS Integrated Circuits", B. Razavi, ISBN: 978-0-07-252493-2
Verwendung des Moduls in weiteren Studiengängen	▪ EL in Bachelor Elektrotechnik 2020 ▪ EL in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ EL in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.18 ES - Eingebettete Systeme

Modulkürzel	ES_BaTIN2020
Modulbezeichnung	Eingebettete Systeme
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	ES - Eingebettete Systeme
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Tobias Krawutschke/Professor Fakultät IME
Dozierende*r	▪ Prof. Dr. Tobias Krawutschke/Professor Fakultät IME ▪ Prof. Dr. Markus Cremer/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt Kompetenzen zur Erstellung eines Eingebetteten Systems (ES) in allen Arbeitsschritten von der Auslegung und Planung des Systems (K.4), der Auswahl der Komponenten (K.8, K.10), der Entwicklung der Software und der Anschaltung an die Anlage / das Gerät (K.5) und seiner prototypischen Inbetriebnahme (K.6). Im parallel laufenden Miniprojekt werden ihre Kompetenzen zur Teamarbeit (K.13), Projektorganisation (K.11) und Kommunikation (durch Präsentationen und Bericht) (K.16) verstärkt.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs/Übungsteil und betreut parallel dazu ein Miniprojekt, in dem die Studierenden ein kleines ES entwickeln.

Wozu: Kompetenzen in der Entwicklung eines ES sind essentiell für technische Informatiker, die im HF 1 arbeiten wollen. Durch die Arbeit an einem Beispielsystem erwerben die Studierenden zudem Erfahrungen, die essentiell für das HF 2 sind, u.a. Anforderungen erfassen, Konzepte zur technischen Lösung entwickeln und diese zu bewerten. Die Durchführung im Team mit dem Dozenten als "Auftraggeber" stärkt die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte

Vorlesung

Entwurfs- und Beschreibungsverfahren
Funktionale Untergliederung
Verhaltensbeschreibung
Objektorientierte Beschreibung
Beschreibung paralleler Abläufe mit Petri-Netzen
Konstruktion eingebetteter Systeme
Hardwareaspekte
Mikrocontroller
SOPC-Lösungen
Anbindung von IO-Bausteinen
Serielle Anbindung
Punkt zu Punkt-Verbindung
Serielle Busse
Parallele Anbindung
DMA
Leistungsverbrauch-Aspekte
Softwareaspekte
Auswahl der Programmiersprache
Assembler
C
C++
andere
SW-Architektur
SingleTask
Zustandsautomat
Statisches Funktionsscheduling
Multitasking
RTOS-basiert
Embedded Linux
Erfüllung von Zeitanforderungen an Tasks
Verteilte eingebetteter Systeme
Grundwissen verteilte Systeme
Schichtenaufbau des Kommunikationssystems
Grundwissen Feldbusse
Grundwissen Internet of Things (IoT)
Programmierung verteilter eingebetteter Systeme

Projekt

Im Team: Entwicklung eines eingebetteten Systems mit einer abgesprochenen Aufgabe, z.B. einer Modellsteuerung eines mechanischen Modells, eines Umweltsensors usw. Projektziel ist ein Prototyp, der die Funktionalität nachweist

Schritte:

1) Beschreibung/Spezifikation

Aufgabenbeschreibung aus Kundensicht im Dialog mit dem Auftraggeber (= Dozent)

Entwicklung eines Konzepts zur Lösung

2) Hardwareauswahl

Recherche geeigneter Bausteine in technischen Handbüchern

3) Modellierung der Lösung

4) Implementierung unter Benutzung von modernen Entwicklungsumgebungen und Programmierstandards, insb. RTOS

komplexe Aufgaben im Team bewältigen

einfache Projekte planen und steuern

Absprachen und Termine einhalten

Präsentation einer Entwicklung

Aufgabenstellung

Projektwischenstand
Ergebnis
Dokumentation in einem Projektbericht
Projektbeschreibung
Umsetzung
Benutzung
Erfahrungen

Übungen

Modellierung eines Eingebetteten Systems gemäß anerkannter Methoden für Reaktive Systeme
Erstellung der Software eines eingebetteten Systems in C
auf Basis einer HAL (Hardware Abstraction Layer) oder unter Benutzung eines RTOS

Lehr- und Lernmethoden	▪ Vorlesung ▪ Projekt ▪ Übungen
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul DR: Grundkenntnisse Technologie und Programmierung von digitaler Logik und Mikrocontrollern ▪ Modul PP: Programmiererfahrung ▪ Modul EG: Grundkenntnisse der Elektrotechnik für die Benutzung von Mikrocontrollern und die Erstellung einfacher Anschaltungen an Mikrocontroller, z.B. mit Spannungsteiler oder einfachem Operationsverstärker ▪ Grundlagen der technischen Informatik Boolesche Logik, Automaten und Schaltwerke Aufbau und Funktionsweise von Mikrocontrollern Mikrocontroller-Programmierung (vorzugsweise in C) Programmiererfahrung mit Entwicklungsumgebungen wie Eclipse
Zwingende Voraussetzungen	Projekt erfordert Anwesenheit im Umfang von: Projektbesprechungen und -präsentation
Empfohlene Literatur	▪ W.Wolff: Computers as Compenents: Principles of Embedded System Design ▪ Wieringa: Design Methods for reactive Systems
Verwendung des Moduls in weiteren Studiengängen	▪ ES in Bachelor Elektrotechnik 2020 ▪ ES in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ ES in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.19 ESP - Eingebettete Systeme - Projekt

Modulkürzel	ESP_BaTIN2020
Modulbezeichnung	Eingebettete Systeme - Projekt
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	ESP - Eingebettete Systeme - Projekt
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Tobias Krawutschke/Professor Fakultät IME
Dozierende*r	Prof. Dr. Tobias Krawutschke/Professor Fakultät IME
Learning Outcome(s)	
Das Modul ESP vertieft die in der Lehrveranstaltung "Eingebettete Systeme" (ES) gewonnenen Kompetenzen zur Konzeption, zum Entwurf und zur Implementierung eingebetteter Systeme in einem Projekt, das über das ganze Semester von einem Team durchgeführt wird. Die Studierenden arbeiten an einer in Absprache mit dem Dozenten ausgewählten Idee für ein ES in den typischen Schritten * Analyse * technische Konzeption unter Benutzung verfügbarer Komponenten * Implementation eines Prototyps Sie präsentieren zu vorgegebenen Terminen ihre Arbeitsschritte und Ergebnisse	
Modulinhalte	
Projekt	
- Entwicklung eines ES - Hardwareauswahl - Recherche geeigneter Bausteine - Lesen von Herstellerdokumenten - Modellierungsverfahren anwenden - Erstellung eines Systemmodells - Verfeinerung der Systemkomponenten - Modellierung der Arbeitsweise - Implementierung - Design der speziellen Komponenten des ES - Inbetriebnahme fertiger Bausteine - Teststrategien entwickeln und umsetzen - Prototypischer Geräteaufbau mit mechanischen/elektronischen Bauteilen - komplexe Aufgaben im Team bewältigen - einfache Projekte planen und steuern - Absprachen und Termine einhalten - Präsentation - Systementwurf - Projektzwischenstand - Ergebnis	
Lehr- und Lernmethoden	Projekt
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung

Workload	150 Stunden
Präsenzzeit	12 Stunden $\triangleq$ 1 SWS
Selbststudium	138 Stunden
Empfohlene Voraussetzungen	▪ Modul ES: Kennen des typischen Aufbaus von ES Anwenden typischer Design- und Programmiermethoden für ES Erstellen eines kleinen ES mit HW- und SW-Anteil ▪ Modul SP: Ablauf eines größeren IT-Projekts Muster der Projektdurchführung mit Meilensteinen ▪ Kurs Eingebettete Systeme
Zwingende Voraussetzungen	Projekt erfordert Anwesenheit im Umfang von: 4 Termine
Empfohlene Literatur	▪ W.Wolff: Computers as Components: Principles of Embedded System Design ▪ Wieringa: Design Methods for reactive Systems ▪ Gessler, Mahr: Hardware/Software Codesign
Verwendung des Moduls in weiteren Studiengängen	ESP in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.20 FIT - Funksysteme für IoT

Modulkürzel	FIT_BaTIN2020
Modulbezeichnung	Funksysteme für IoT
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	FIT - Funksysteme für das IoT
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Uwe Dettmar/Professor Fakultät IME
Dozierende*r	Prof. Dr. Uwe Dettmar/Professor Fakultät IME

Learning Outcome(s)

Was? Erlernen der Nutzung bestehender Funktechnologien zur Planung und Entwicklung von neuen Anwendungen im Mobilfunk und im Bereich des IoT (smart home, smart city, smart grid, smart farming etc.) sowie der Digitalisierung der Industrie (Industrie 4.0).
Womit? durch Beschäftigung mit existierenden und neu auf den Markt kommenden geeigneten funktechnischen Standards und Geräten und deren Grundlagen
Wozu? zum Entwurf, der anwendungsspezifischen Auswahl und Beurteilung von innovativen Lösungen zur Messwertaufnahme, sicheren Datenübertragung und Steuerung von Prozessen in den Bereichen des Internet-of-Things und von Industrie 4.0.

Modulinhalte

Vorlesung / Übungen

Vorlesung und Übungen werden in der Lehrveranstaltung kombiniert. Nach Vorstellung von neuem Lernstoff durch den Dozenten in Form von kurzen Blöcken wird dieser direkt von den Studierenden durch kurze Matlab- und Python-Übungen angewendet und vertieft. Längere Übungsaufgaben werden bereits zu Hause vorbereitet und die verschiedenen Lösungsvorschläge in der Präsenzveranstaltung besprochen.

Kenntnisse zu folgenden Themen werden vermittelt:

- Einführung, Was ist IoT/ Industrie 4.0?
- Überblick: Märkte und Einsatzgebiete für drahtlose Kommunikation
- Standards, Grundlagen zur drahtlosen Übertragungstechnik
- Sensoren, Aktoren und uC
- Vielfachzugriff und Datensicherung in Sensornetzen
- Techniken für höhere Datenraten (OFDM, MIMO etc.)
- Network, Fog und Cloud Computing
- Standards in Mobilfunk (4G, 5G), WLAN, LPWAN, WMAN und WPAN

Die Studierenden lernen die o.g. Themen in der Vorlesung kennen, erwerben Grundwissen und vertiefen dieses durch Selbststudium mit Hilfe von Literatur, YouTube Videos und anderen Netzressourcen (selbstständige Informationsbeschaffung), sowie in Lerngruppen (Teamwork).

Durch die Kombination mit kleinen Übungsaufgaben und Programmen wird in der Präsenzveranstaltung bereits ein aktiver Umgang mit den vorgestellten Verfahren befördert. Umfangreichere Rechenaufgaben werden am Ende der Veranstaltung behandelt und die Lösungswege diskutiert, um dadurch den Studierenden relevante Problemstellungen vorzustellen und ihre Fähigkeit zur Lösungsfindung zu entwickeln.

Die Studierenden lernen darüber hinaus:

- nachrichtentechnische System zu analysieren und deren Performanz zu ermitteln bzw. abzuschätzen.
- Geeignete Standards für spezifische Anwendungen auszuwählen
- Kenntnisse auf technische Problemstellungen anzuwenden

Projekt

In Kleingruppen bearbeiten die Studierenden Projekte aus dem Bereich des IoT. Dabei verwenden Sie HW oder SW, um aktuelle Funkstandards zu untersuchen oder anzuwenden, Daten z.B. von Sensoren aufzunehmen, zu sammeln, darzustellen und auszuwerten. Die Arbeiten verschiedener Kleingruppen können zu einem Gesamtprojekt kombiniert werden.

Die Ergebnisse werden in einer Präsentation vorgestellt und bewertet. Sie können mit bis zu 30% in die Abschlussnote eingehen.

Seminar

Alternativ: Ausgabe einer Seminararbeit zu einem aktuellen Thema aus der Veranstaltung. Abgabe und Bewertung der Arbeit.

Eigenständige Literaturrecherche, Analyse der Quellen, themengerechte, verständliche Darstellung, Diskussion und Bewertung der gefundenen Ergebnisse.

Praktikum

Alternativ: Durchführung von vorgegebenen Praktikumsaufgaben, die Aspekte der Vorlesung vertiefen. Dies kann HW oder SW basiert erfolgen.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Projekt ▪ Seminar ▪ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	68 Stunden $\pm$ 6 SWS
Selbststudium	82 Stunden
Empfohlene Voraussetzungen	▪ Modul SV: Bandbreitedefinition, Denken im Frequenzbereich, Fourier Transformation, Signale und Systeme Diskrete Fourier Transformation ▪ Die Studierenden sollten Grundkenntnisse der digitalen Kommunikationstechnik und der Signaltheorie besitzen. Sie sollten vertraut sein mit Standardprotokollen zur Datenübertragung und dem OSI Schichtenmodell. Sie sollten insbesondere Grundkenntnisse zur physikalischen Schicht und der Sicherungsschicht mitbringen. Aus der Mathematik wird Grundwissen in den Bereichen Lineare Algebra und Statistik benötigt. Grundlegende Programmierkenntnisse runden das Anforderungsprofil ab.
Zwingende Voraussetzungen	▪ Projekt erfordert Anwesenheit im Umfang von: 4 Termine ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Projekt
Empfohlene Literatur	▪ DAHLMAN, E. ; PARKVALL, S. ; SKÖLD, J. : 5G NR : the next generation wireless access technology. 1st. Elsevier Science, 2018 ▪ FINKENZELLER, K. : RFID Handbuch. Hanser, 2008. ▪ FÖRSTER, A. : Introduction to Wireless Sensor Networks. Wiley-IEEE Press, 2016. ▪ GEIER, J. : Designing and deploying 802.11 wireless networks, Cisco Press, 2015. ▪ LIAO, R. ; BELLALTA, B. ; OLIVER, M. ; NIU, Z. : MU-MIMO MAC Protocols for Wireless Local Area Networks: A Survey. In: IEEE Commun. Surv. Tutorials 18 (2016) ▪ Mobile positioning and tracking : from conventional to cooperative techniques. Wiley-IEEE Press ▪ TANENBAUM, A. S. ; WETHERALL, D. : Computer networks. Pearson Education, 2014
Verwendung des Moduls in weiteren Studiengängen	▪ FIT in Bachelor Elektrotechnik 2020 ▪ FIT in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ FIT in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.21 FSA - Formale Sprachen und Automatentheorie

Modulkürzel	FSA_BaTIN2020
Modulbezeichnung	Formale Sprachen und Automatentheorie
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	FSA - Formale Sprachen und Automatentheorie
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	2
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Hans Nissen/Professor Fakultät IME
Dozierende*r	Prof. Dr. Hans Nissen/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt verschiedene Kompetenzen für die Analyse, die Formalisierung und den Umgang mit formalen, abstrakten Strukturen in informationstechnischen Systemen:
 Formalisierung und Analyse von Systemen aus abstrakter Perspektive (K.1, K.2, K.3, K.4, K.5),
 Formalisierung einer gegebenen Struktur als formale Sprache (K.2, K.3),
 Spezifikation einer Grammatik zu einer gegebenen Sprache (K.2, K.3),
 Ableitung von akzeptierenden Automaten für gegebene Sprachen (K.1, K.2, K.5),
 Transformation einer Beschreibungsform einer formalen Sprachen in eine andere, äquivalente Beschreibungsform (K.1, K.2, K.4),
 Analyse einer Struktur bezüglich ihrer Komplexitäts-Eigenschaften (K.2, K.4, K.12).

Womit: Der Dozent vermittelt Wissen und Fertigkeiten in einem Vorlesungs/Übungsteil unter Verwendung einer Vielzahl von abstrakten und praktischen Beispielen.
 Die Analyse und die Formalisierung abstrakter Strukturen werden in der Vorlesung und der Übung intensiv geübt.

Wozu: Bei der Entwicklung von Software und Systemen werden sehr oft formale Strukturen, wie beispielsweise Automaten und Grammatiken, betrachtet.
 Diese Strukturen abstrahieren von der realen Welt und überführen insbesondere das dort beobachtbare bzw. gewünschte Verhalten in ein für die Implementierung geeignetes formalisiertes Modell.
 Für eine erfolgreiche Tätigkeit in der System- und Softwareentwicklung ist die sichere Abstraktion und Formalisierung von Strukturen und Verhalten aus der realen Welt erforderlich. Denn nur mit einer geeigneten Formalisierung ist die korrekte Verarbeitung von Informationsstrukturen und die korrekte Umsetzung von technischen Anwendung (z.B. zur Steuerung von Anlagen) möglich (HF.1).
 Diese Lehrveranstaltung liefert weiterhin die Kompetenzen für die Analyse von Systemen und den zugrundeliegenden Konzepten bezüglich ihrer Eignung zur Verarbeitung der geforderten Strukturen bzw. der Sicherstellung des gewünschten Verhaltens (HF.2).

Modulinhalte**Vorlesung / Übungen**

Formale Sprachen und Chomsky-Hierarchie
 Formalisierung von Grammatiken
 Formalisierung von abstrakten Rechnermodellen
 verschiedene endliche Automaten
 Kellerautomat
 Turingmaschine
 reguläre Ausdrücke
 Eigenschaften unterschiedlicher Sprachklassen
 Abgeschlossenheit
 Entscheidbarkeit
 Pumping Lemma
 Sprachklasse einer gegebenen Sprache bestimmen
 formale Sprachen spezifizieren
 Grammatik für gegebene Sprache erstellen
 Automat für gegebene Sprache erstellen
 Automat für gegebene Grammatik erstellen
 Formalisierungen transformieren
 formale Beweise zu formalen Sprachen, Grammatiken und Automaten durchführen
 Probleme der realen Welt formalisieren
 abstrakte Automaten für reale Probleme entwerfen

Lehr- und Lernmethoden Vorlesung / Übungen

Prüfungsformen mit Gewichtung siehe Prüfungsordnung

Workload 150 Stunden

Präsenzzeit 34 Stunden $\pm$ 3 SWS

Selbststudium 116 Stunden

Empfohlene Voraussetzungen keine

Zwingende Voraussetzungen

Empfohlene Literatur

- Uwe Schöning: Theoretische Informatik - kurzgefasst, Spektrum Akademischer Verlag, 5. Auflage, 2008
- Rolf Socher: Theoretische Grundlagen der Informatik Carl Hanser Verlag, 2007
- Gottfried Vossen, Kurt-Ulrich Witt: Grundkurs Theoretische Informatik 4. Auflage, Vieweg Verlag, 2006
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie 3. Auflage, Pearson Studium, 2011

Verwendung des Moduls in weiteren Studiengängen FSA in Bachelor Informatik und Systems-Engineering 2024

Besonderheiten und Hinweise

Letzte Aktualisierung 19.7.2025, 14:32:16

6.22 GRT - Graphentheorie

Modulkürzel	GRT_BaTIN2020
Modulbezeichnung	Graphentheorie
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	GRT - Graphentheorie
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Hubert Randerath/Professor Fakultät IME
Dozierende*r	
Learning Outcome(s)	▪ (WAS) Studierende können bestimmte Problemstellungen der Informatik abstrakt modellieren und auf Basis dieser Modelle gängige und wohlverstandene Methoden und Algorithmen zur Lösung anwenden ▪ (WOMIT) indem sie ▪ die Problemstellung als Graph darstellen, ▪ die Graphen klassifizieren und anwendbare Algorithmen identifizieren und ▪ diese Algorithmen implementieren, ▪ (WOZU) damit sie effektive und effiziente Lösungen für wiederkehrende Anforderungen wie beispielsweise Wegfindungen, Optimierung von (Netzwerk-)Routing oder Ermittlungen von Abhängigkeiten in Programmen realisieren können.

Modulinhalte**Vorlesung**

- Grundlagen der Graphentheorie
 - Definition Graph
 - vollständige und bipartite Graphen
 - Isomorphie von Graphen
 - Adjazenz- und Inzidenzmatrix
 - Wege und Kreise
 - Zusammenhang von Graphen
 - gerichtete Graphen
 - Wälder
 - Bäume
 - elementare Graphenparameter
- Traversierung in Graphen und das kürzeste Wege Problem
 - Breitensuche
 - Tiefensuche
 - Topologische Sortierung
 - Eulertouren
 - Hamiltonkreise
 - Dijkstra-Algorithmus
 - Bellman-Ford-Algorithmus.
- Matchings und Flüsse
 - Matchings
 - Ungarische Methode
- Färbungen
 - Knotenfärbungen
 - Kantenfärbungen
 - Listenfärbungen
 - Turniere und Spielpläne
 - Planare Graphen
 - Vier-Farben-Satz
 - Perfekte Graphen
 - Färbungsalgorithmen

Übungen

- Die Studierenden lösen graphentheoretische und graphenalgorithmische Probleme mit Maple

translation not found

Lehr- und Lernmethoden	▪ Vorlesung
	▪ Übungen
	▪ translation not found

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	NaN Stunden $\triangleq$ NaN SWS
--------------------	----------------------------------

Selbststudium	NaN Stunden
----------------------	-------------

Empfohlene Voraussetzungen

Zwingende Voraussetzungen	Übungen erfordert Anwesenheit im Umfang von: 2 Termine
----------------------------------	--

Empfohlene Literatur	▪ Diskrete Mathematik, M. Aigner, Vieweg-Verlag ▪ Graphentheorie, R. Diestel, Springer Verlag ▪ Graphentheorie, P. Tittman, Fachbuchverlag Leipzig ▪ Graphen an allen Eckern und Kanten, L. Volkmann, RWTH Aachen
-----------------------------	---

Verwendung des Moduls in weiteren Studiengängen	GRT in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.23 GSP - Grundlagen der Systemprogrammierung

Modulkürzel	GSP_BaTIN2020
Modulbezeichnung	Grundlagen der Systemprogrammierung
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	GSP - Grundlagen der Systemprogrammierung
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	2
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Tobias Krawutschke/Professor Fakultät IME
Dozierende*r	- Prof. Dr. Tobias Krawutschke/Professor Fakultät IME - Norbert Kellersohn/Lehrkraft für besondere Aufgaben

Learning Outcome(s)

Was: Das Modul vermittelt die Kompetenzen zur Verwendung von Mikrocontrollersystemen (Hardware inkl. Echtzeitbetriebssystem) für die Erarbeitung von Problemlösungen aus dem Bereich Messen-Steuern-Regeln. Darüber hinaus werden die Studierenden in die Lage versetzt eigene Treiber für vom Betriebssystem nicht unterstützte Hardwarekomponenten zu entwickeln und zu nutzen. Alle Implementierungen erfolgen in der Programmiersprache C, wodurch die Einführung in diese Programmiersprache ein integraler Bestandteil dieses Moduls ist. Die Studierenden werden in die Lage versetzt, fachspezifische Begriffe, Tools und Techniken im praktischen Umfeld sicher anzuwenden. Aufbauend auf den in der Vorlesung vermittelten Kenntnissen werden komplexere Problemstellungen analysiert, auf Teilsysteme heruntergebrochen und modelliert. Darauf aufbauend wird die Problemlösung mittels Entwurfswerkzeugen implementiert und am Zielsystem in Betrieb genommen.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. Im Praktikum erarbeiten die Studierenden in Kleingruppen Problemlösungen und verteidigen diese.

Wozu: Kompetenzen in der Verwendung von Mikrocontrollern nebst Echtzeitbetriebssystem sind essentiell für technische Informatiker, die im HF 1 arbeiten wollen. Durch die Entwicklung von Problemlösungen erwerben die Studierenden zudem Erfahrungen, die essentiell für das HF 2 sind. Eine projektorientierte Durchführung der Praktika in kleinen Teams mit dem Dozenten als "Auftraggeber" initiiert die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte**Vorlesung / Übungen**

- Grundlagen der C-Programmierung: Datentypen, Ausdrücke, Strukturen, Präprozessor, Zeiger, Funktionen,
- Werkzeuge zur Programmierung: Compiler, Linker, Debugger, Simulator/Emulator
- Hardwarenahe I/O-Programmierung in C: Port Aufbau, I/O Adressierungsarten, Treiberbibliotheken, Bitmanipulationen
- Programmierung von Aufgaben des Messens, Steuern und Regels in C: Realisierung von Moore- und Mealy-Automaten in C
- Aufbau und Funktionsweise eines Echtzeitbetriebssystems: Anforderungen, Vergleich mit herkömmlichen OS, Multitasking, Priorität, Taskzustände, Mutex, Semaphore, Task Kommunikation, Deadlocks, Race-Conditions
- I/O-Schnittstellen eines Rechnersystems und deren Nutzung mittels C am Beispiel des dedizierten Kleinrechnersystems:
- Interrupts: Quellen, Arten, Vektor Tabelle, Service Routine, Timing, Priorisierung, Hardware Quellen
- Laufzeitsystem für C-Programme:
 - Stackverarbeitung, Unterprogramme, Parameterübergabe
 - Speicherverwaltung, Kontext-Speicherung

Praktikum

- zielgerichtetes Handhaben der Software-Entwicklungsumgebung
- komplexe Aufgaben in einem Kleinteam bewältigen
- Verstehen und erläutern der Arbeitsweise eines Mikrocontroller-System (Hardware und Echtzeitbetriebssystem)
- Erstellen von Treiberbibliotheken in C für verschiedene I/O-Schnittstellen mit Unterstützung ihrer Interruptfähigkeit
- Systemverhalten aus spezifizierenden Texten herleiten
- Erarbeitung von Problemlösungen aus dem Bereich Messen-Steuern-Regeln, die sich mit C-Programmen realisieren lassen
- Laufzeitsystem für C-Programme analysieren und beschreiben

Lehr- und Lernmethoden	▪ Vorlesung / Übungen
	▪ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	grundlegende prozedurale Programmierkenntnisse Grundlegende Funktionsweise eines Von-Neumann-Rechners Grundlagen der Digitaltechnik Automaten u. Zustandsüberführungsdiagramme
-----------------------------------	---

Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: 2 Termine
----------------------------------	--

Empfohlene Literatur	▪ Martini: Rechnerarchitektur, Fachbuchverlag Leipzig (Carl Hanser) ▪ Oberschelp/Vossen: Rechneraufbau und Rechnerstrukturen, Oldenbourg Verlag ▪ Vogt, C: C für Java-Programmierer ▪ Tanenbaum, Goodman: Computerarchitektur, Pearson Studium (Prentice Hall)
-----------------------------	---

Verwendung des Moduls in weiteren Studiengängen	GSP in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.24 GUI - Graphische Oberflächen und Interaktion

Modulkürzel	GUI_BaTIN2020
Modulbezeichnung	Graphische Oberflächen und Interaktion
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	GUI - Graphische Oberflächen und Interaktion
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	3
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Andreas Behrend/Professor Fakultät IME
Dozierende*r	Ali Shafieian/wissenschaftlicher Mitarbeiter Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt Kompetenzen zur Programmierung von grafischen Benutzeroberflächen. In praktischer Arbeit analysieren die Studierenden Problemstellungen (K2), implementieren Lösungen mit Hilfe von Standardwerkzeugen (K6, K9) und prüfen sie (K7). Sie recherchieren dazu in Online-Dokumentationen (K8, K15) und passen vorhandene Software an (K10).

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. In den Übungen und insbesondere im Praktikum arbeiten die Studierenden in Kleingruppen und verteidigen ihre Lösungen (K8, K13).

Wozu: Grafische Benutzeroberflächen spielen im Programmierumfeld eine zentrale Rolle und somit auch Kenntnisse, sie zu programmieren und in bestehende Systeme zu integrieren (HF1). Durch ihre praktische Programmierarbeit erwerben die Studierenden zudem weitere Erfahrungen, die wichtig sind für die Erfassung von Anforderungen, die Entwicklung von Konzepten zur technischen Lösung und zu ihrer Bewertung (HF2). Die Durchführung im Team mit dem Dozenten als "Auftraggeber" stärkt die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte

Vorlesung / Übungen

grundlegende Begriffe und Techniken der Erstellung graphischer Benutzeroberflächen unter Microsoft Windows
Fensterkonzept
Nachrichtenkonzept

Anwenden der durch das Betriebssystem gegebenen API in Java
Einbinden der Pakete AWT und Swing
Diskussion der Vor- und Nachteile

Umgang mit der Microsoft Windows API

Umgang mit AWT- und Swing-Klassen zur Erstellung von graphischen Benutzeroberflächen in Java

Praktikum

Programmierung von Graphischen Benutzeroberflächen am Beispiel individueller Aufgaben - Details wie unter "Vorlesung/Übung" angegeben

Umgang mit Softwareentwicklungsumgebungen für GUIs

Realisierung von GUI-Applikationen mittlerer Komplexität in kleinen Teams

Lehr- und Lernmethoden	■ Vorlesung / Übungen ■ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\triangleq$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul PI2: Sicherer Umgang mit einer objektorientierten Programmiersprache. ▪ Modul AD: Tieferes Verständnis der Verwendung von C++ und Java bei der Analyse von Algorithmen ▪ Prozedurale und objektorientierte Programmierung in C/C++ und Java Struktur und Funktionalität von Betriebssystemen
Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: 3 Termine
Empfohlene Literatur	▪ keine
Verwendung des Moduls in weiteren Studiengängen	GUI in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.25 HF - Hochfrequenztechnik

Modulkürzel	HF_BaTIN2020
Modulbezeichnung	Hochfrequenztechnik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	HF - Hochfrequenztechnik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Rainer Kronberger/Professor Fakultät IME
Dozierende*r	Prof. Dr. Rainer Kronberger/Professor Fakultät IME

Learning Outcome(s)

Die Studierenden können hochfrequenztechnische Probleme lösen, indem sie hierfür geeignete spezielle Methoden der Elektrotechnik anwenden, um später hochfrequenztechnische Schaltungen, Baugruppen zu analysieren, entwickeln und herzustellen.

Die Studierenden können hochfrequenztechnische Schaltungen entwickeln, indem sie hierfür geeignete spezielle Methoden der Elektrotechnik und Elektronik anwenden, um später hochfrequenztechnische Systeme zu entwickeln und herzustellen.

Modulinhalte

Vorlesung

Die Studierenden lernen die Besonderheiten und Unterschiede elektrotechnischer Grundprinzipien, Vorgänge und Schaltungen bei hohen und höchsten Frequenzen.

Es werden theoretischen Grundlagen in Verbindung mit praktischen Anwendungsbeispielen der Hochfrequenztechnik vermittelt und der Unterschied zur konventionellen Elektrotechnik wird erklärt und geschult. Im Praktikum lernen die Studierenden grundlegende Messverfahren und -geräte der Hochfrequenztechnik kennen.

Vorlesungs- und Übungsbegleitend wird die Anwendung eines professionelles HF-Simulationsprogramm trainiert, das allen Studierenden im Labor und zu Hause zur Verfügung steht.

- Lineare, passive Transformationsschaltungen mit L und C
- Streuparameter und Streumatrizen
- Leitungstheorie, Leitungsschaltungen, Leitungstransformationen
- Resonanzschaltungen und Filterschaltungen
- Hochfrequenzmaterialeigenschaften

Übungen / Praktikum

Begleitende Übung und begleitendes Praktikum zur Vorlesung

- Lineare, passive Transformationsschaltungen mit L und C
- Streuparameter und Streumatrizen
- Leitungstheorie, Leitungsschaltungen, Leitungstransformationen
- Resonanzschaltungen und Filterschaltungen
- Hochfrequenzmaterialeigenschaften

Die Studierenden lernen die Besonderheiten und Unterschiede elektrotechnischer Grundprinzipien, Vorgänge und Schaltungen bei hohen und höchsten Frequenzen.

Es werden theoretischen Grundlagen in Verbindung mit praktischen Anwendungsbeispielen der Hochfrequenztechnik vermittelt und der Unterschied zur konventionellen Elektrotechnik wird erklärt und geschult. Im Praktikum lernen die Studierenden grundlegende Messverfahren und -geräte der Hochfrequenztechnik kennen.

Vorlesungs- und Übungsbegleitend wird die Anwendung eines professionelles HF-Simulationsprogramm trainiert, das allen Studierenden im Labor und zu Hause zur Verfügung steht.

Lehr- und Lernmethoden	▪ Vorlesung ▪ Übungen / Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul EG: Grundlegende Kenntnisse der Elektrotechnik Grundlegende Kenntnisse der Wechselstromtechnik Grundlegende Kenntnisse zu elektrischen und magnetischen Feldern ▪ Modul MA1: Grundlegende Kenntnisse der Mathematik ▪ GE1-GE3, MA1, MA2
Zwingende Voraussetzungen	Übungen / Praktikum erfordert Anwesenheit im Umfang von: Praktikumstermine
Empfohlene Literatur	▪ Meinke/ Gundlach: Taschenbuch der Hochfrequenztechnik Bd. 1-3 Springer Verlag Zinke/ Brunswig: Hochfrequenztechnik 1, Filter, Leitungen, Antennen, Springer Verlag Detlefsen/Siart: Grundlagen der HF-Technik. Oldenbourg Verlag
Verwendung des Moduls in weiteren Studiengängen	▪ HF in Bachelor Elektrotechnik 2020 ▪ HF in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ HF in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.26 IAK - Ingenieurakustik

Modulkürzel	IAK_BaTIN2020
Modulbezeichnung	Ingenieurakustik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	IAK - Ingenieurakustik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr.-Ing. Christoph Pörschmann/Professor Fakultät IME
Dozierende*r	Prof. Dr.-Ing. Christoph Pörschmann/Professor Fakultät IME

Learning Outcome(s)

Was: Durch das Modul lernen die Studierenden die grundlegenden Konzepte und physikalischen Zusammenhänge der Akustik kennen werden in die Lage versetzt, diese zu beschreiben, zu analysieren und die Auswirkungen veränderter Einflussgrößen abzuschätzen.

Womit: Durch das Verständnis und die Anwendung der in der Vorlesung präsentierten Grundlagen erlernen die Studierenden, wie sich Schall ausbreitet, wie er erzeugt wird und welche physikalischen Phänomene dabei eine Rolle spielen. Ein weiteres Verständnis der grundlegenden Zusammenhänge wird durch das Praktikum bewirkt, in dem die Studierenden selbst Messungen vornehmen und relevante Parameter bestimmen. Sie erlernen somit, die physikalischen Zusammenhänge zu den entsprechenden Modellen und Kennziffern in Beziehung setzen.

Wozu: Akustische Zusammenhänge spielen im Alltag eines Ingenieurs an vielen Stellen eine wesentliche Rolle, vom Lärmschutz, über Grundprinzipien der Schallausbreitung in Räumen. Für medientechnische Systeme und Medienprodukte spielt die gezielte Anregung und kontrollierte Ausbreitung von Schall eine große Rolle. Die Veranstaltung vermittelt hierzu die nötigen Grundkenntnisse und Aufbaukenntnisse.

Modulinhalte**Vorlesung / Übungen**

Einführung der akustischen Grundgrößen
 Schalldruck, Schallschnelle, Schallfluss Schalleistung
 Logarithmische Größen und Pegel

Mechanische und akustische Schwingungssysteme
 Mechanische Schwingungssysteme
 Akustische Schwingungssysteme

Schallausbreitung im Raum
 Homogene ebene Welle
 stehende Wellen
 Resonanzsysteme
 Beugung, Brechung, Reflexion

Punktschallquellen
 Verhalten von Schalldruck und Schallschnelle
 Elementarstrahlersynthese

Schallwandler (Lautsprecher und Mikrophone)
 Prinzipien der Richtmikrophone
 Elektrodynamische Mikrophone und Kopfhörer
 Piezoelektrische Mikrophone und Kopfhörer
 Dielektrische Mikrophone

Absorber
 Poröse Absorber
 Helmholtz Resonatoren als Absorber
 Plattenabsorber

Analyse und Beschreibung von Systemen mit Lautsprechern und Mikrofonen

Berechnung und Beschreibung der gesamten Kette der Schallausbreitung vom Mikrophon über die mechanoelektrische Wandlung, die Weiterleitung über eine Nachrichtenstrecke sowie die Umwandlung über einen elektromechanischen Wandler und die Schallabstrahlung

Praktikum

Simulation der Ausbreitung von Schallwellen
 Untersuchungen der Schallreflexion am Kundt'schen Rohr
 Analyse von Eigenmoden
 textlich beschriebene Aufgaben in praktische Messungen umsetzen
 funktionstüchtige Messaufbauten erstellen
 fachgerechte Dokumentationen für durchgeführte Messungen anfertigen
 Messergebnisse bewerten und diskutieren

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	Grundkenntnisse Mechanik Kenntnisse Zeit- und Frequenzbereich Komplexe Rechnung Grundkenntnisse Integral- und Differentialrechnung
-----------------------------------	---

Zwingende Voraussetzungen	■ Praktikum erfordert Anwesenheit im Umfang von: 3 Praktikumstermine und 1 Hörversuch ■ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	■ Boré, G., Peus, S. (1999). „Mikrophone für Studio und Heimstudio-Anwendungen – Arbeitsweise und Ausführungsbeispiele,“ Hrsg. Georg Neumann GmbH, Berlin. ■ Blauert, J., Xiang, N. (2008). „Acoustic for Engineers – Troy Lectures,“ Springer Verlag, Heidelberg. ■ Blauert, J., Braasch, J., Jekosch, U. (2012). „Acoustics for Communication – Dresden Lectures,“ Springer Verlag Heidelberg, in Vorbereitung. ■ Dickreiter, M., Hoeg, W., Dittel, V., Wöhr, M. (2008). „Handbuch der Tonstudioteknik,“ 7. Auflage, Saur Verlag, München. ■ Görne, T. (2011). „Tontechnik,“ Hanser Verlag München. ■ Kuttruff, H. (2004). „Akustik – Eine Einführung,“ S. Hirzel Verlag, Stuttgart. ■ Cremer, L. (1976). „Vorlesungen über Technische Akustik,“ Springer Verlag, Berlin, Heidelberg. ■ Lord Rayleigh (1896). „The Theory of Sound,“ 2nd Edition 1896, Dover Publ. New York. ■ Müller, G., Möser, M. (2004). „Taschenbuch der Technischen Akustik,“ Springer Verlag Berlin, 3. Auflage. ■ Veit, I. (2005). „Technische Akustik“, Kamprath-Reihe, Vogel-Verlag, Würzburg. ■ Weinzierl, Stefan (2008). „Handbuch der Audiotechnik,“ Springer Verlag, Berlin. ■ Blauert, J., (2005) „Communication Acoustics,“ Springer Verlag Heidelberg, ■ Blauert, J., (2021) „Acoustics for Communication,“ Springer Verlag Heidelberg, upcoming
Verwendung des Moduls in weiteren Studiengängen	■ IAK in Bachelor Elektrotechnik 2020 ■ IAK in Bachelor Elektrotechnik und Informationstechnik 2024 ■ IAK in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.27 IBV - Industrielle Bildverarbeitung

Modulkürzel	IBV_BaTIN2020
Modulbezeichnung	Industrielle Bildverarbeitung
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	BV1 - Bildverarbeitung
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Jan Salmen/Professor Fakultät IME
Dozierende*r	Prof. Dr. Jan Salmen/Professor Fakultät IME

Learning Outcome(s)

Was: Die Studierenden erlernen den grundlegenden Aufbau und den software-technischen Zugriff auf digitale Bilder sowie die Methoden und Verfahren der digitalen Bildverarbeitung zur Bildverbesserung, Farbbildverarbeitung und Vermessung von Bildinhalten. Sie erlernen eine komplexere Problemstellung aus dem Bereich der Bildverarbeitung zu analysieren, auf Teilsysteme herunter zu brechen und geeignete Methoden und Verfahren auszuwählen. Darauf aufbauend wird die Problemlösung mittels Entwurfswerkzeugen implementiert und validiert.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. Im Praktikum erarbeiten die Studierenden in Kleingruppen Problemlösungen und verteidigen diese.

Wozu: Die Studierenden werden in dem Modul befähigt, im industriellen Umfeld Problemlösung berührungsloser Inspektions- und Meßaufgaben mittels Bildverarbeiten zu lösen und bestehende Lösungen zu bewerten.

Modulinhalte

Vorlesung

- | Digitale Bilder
- | Punktoperationen (z.B. Gamma-Korrektur)
- | Histogramme, Bildverbesserung
- | Umwandlung von Bildformaten: Farbbilder, Grauwertbilder, Binärbilder
- | Morphologische Operatoren
- | Segmentierung, Regionen in Binärbildern und ihre Eigenschaften
- | Lineare Filter, insbesondere Kantenfilter
- | Weitere Farbräume, Clustering und Klassifikation von Farben
- | Dithering
- | Finden einfacher Formen: Hough-Transformation und RANSAC
- | Ähnlichkeit von Bildern, Template Matching

Praktikum

- | Einführung in Python
- | Verfahren zur Bildverarbeitung implementieren und testen, z.B. Bildverbesserung
- | Verfahren kombinieren, um praktische Anwendungsfälle zu lösen, z.B. Green Screen ersetzen oder Foto-Mosaik erstellen
- | Einfache Objekterkennung realisieren, im Wesentlichen ohne maschinelles Lernen

Lehr- und Lernmethoden	▪ Vorlesung ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	34 Stunden $\pm$ 3 SWS
Selbststudium	116 Stunden
Empfohlene Voraussetzungen	▪ Modul PI1: Entwickeln von Programmen zur Lösung konkreter Problemstellungen, abstrahieren von Problembeschreibungen in Algorithmen und überprüfen von Programmen auf Fehler. ▪ Modul MA1: Problemlösungskompetenz aus dem Bereich lineare Algebra und der Analysis. Sicheres Beherrschen der entsprechenden Symbole und Formalismen ▪ Modul SV: Darstellung von zeitdiskreten Signalen im Zeit und Frequenzbereich (DFT). ▪ Grundstudium Mathematik ▪ Grundstudium Informatik ▪ Grundstudium Signaltheorie
Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: 4 Fachgespräche
Empfohlene Literatur	▪ Burger/Burge: Digitale Bildverarbeitung
Verwendung des Moduls in weiteren Studiengängen	▪ IBV in Bachelor Elektrotechnik 2020 ▪ BV in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ BV1 in Bachelor Medientechnologie 2020 ▪ BV in Bachelor Medientechnologie 2024 ▪ BV in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.28 IOT - IoT Protokolle und Anwendungen

Modulkürzel	IOT_BaTIN2020
Modulbezeichnung	IoT Protokolle und Anwendungen
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	IOT - IoT Protokolle und Anwendungen
ECTS credits	5
Sprache	deutsch und englisch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Harald Elders-Boll/Professor Fakultät IME
Dozierende*r	Prof. Dr. Harald Elders-Boll/Professor Fakultät IME

Learning Outcome(s)

Kennen und Anwenden die wichtigsten Protokolle, Anwendungen und Datenanalyse und Sicherheitstechniken für das Internet der Dinge (IoT) sowie der Digitalisierung der Industrie (Industrie 4.0) durch Vermittlung der zugrundeliegenden Methoden und Konzepte und deren Anwendung in Praktikumsaufgaben zur Vernetzung und Sicherheit von IoT-Endgeräten, zum Entwurf, der anwendungsspezifischen Auswahl und Beurteilung von innovativen und sicheren Anwendungen in den Bereichen des Internet of Things und der Industrie 4.0.

Modulinhalte

Vorlesung / Übungen

Vorlesung und Übungen werden in der Lehrveranstaltung kombiniert. Nach Vorstellung von neuem Lernstoff durch den Dozenten in Form von kurzen Blöcken wird dieser direkt von den Studierenden durch kurze Übungen angewendet und vertieft. Längere Übungsaufgaben werden bereits zu Hause vorbereitet und die verschiedenen Lösungsvorschläge in der Präsenzveranstaltung besprochen.

Kenntnisse zu folgenden Themen werden vermittelt:

Einführung in des Internet der Dinge
 IoT Anwendungsfelder
 Hard- und Softwaregrundlagen des IoT
 IoT Systeme und Architekturen
 IoT Kommunikationsprotokolle
 IoT Protokolle der Anwendungsschicht (MQTT, CoAP, HTTP, REST)
 Datenanalyse und maschinelles Lernen für IoT
 IoT Sicherheit

IoT Architekturen unterscheiden können. IoT Systeme unter Einsatz geeigneter Tools analysieren. IoT Endgeräte in IoT Systeme einbinden. Sicherheit von IoT Systemen abschätzen und analysieren.

Praktikum

Sensoren und Aktoren an Mikroprozessoren und Einplatinenrechner anbinden
 Netzwerkverbindung von IoT Endgeräten herstellen
 Messwerte in die Cloud übertragen
 Hard- und Software von IoT Endgeräten kompromittieren
 Kommunikation von IoT Geräten abhören

Lehr- und Lernmethoden	■ Vorlesung / Übungen ■ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul NP: Grundlagen von Rechnernetzen Netzwerkanwendungen und Protokolle Grundlagen der Transportschicht Adressierung und Routing Grundlagen der Sicherungsschicht Grundlagen der Netzwerksicherheit ▪ Grundlagen von Rechnernetzen Netzwerkanwendungen und Protokolle Grundlagen der Transportschicht Adressierung und Routing Grundlagen der Sicherungsschicht Grundlagen der Netzwerksicherheit
Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 4 Termine ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	▪ P. Lea, "Internet of Things for Architects", Pakt, 2018 ▪ A. Bahga, V. Madisetti, "Internet of Things A Hands-on Approach", Bagha & Madisetti ▪ B. Adyan, D. Obermaier, P. Fremantle, "The Technical Foundations of IoT", Artech House, 2017
Verwendung des Moduls in weiteren Studiengängen	▪ IOT in Bachelor Elektrotechnik 2020 ▪ IOT in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ IOT in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.29 IPM - IT-Projektmanagement

Modulkürzel	IPM_BaTIN2020
Modulbezeichnung	IT-Projektmanagement
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	IPM - IT-Projektmanagement
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Chunrong Yuan/Professor Fakultät IME
Dozierende*r	▪ Prof. Dr. Chunrong Yuan/Professor Fakultät IME ▪ Dr. Nikolaus Schmidt/Lehrbeauftragter

Learning Outcome(s)

Was: Das Modul vermittelt Kompetenzen zur Organisation und Verwaltung von IT Projekten in allen relevanten Arbeitsschritten von der Anforderungsanalyse (K.1), Projektplanung und Aufwandschätzung (K.4), Risikomanagement und Personalführung (K.9), Qualitätsmanagement und Fortschrittskontrolle (K.11), bis zu Projektabschluss, so dass die Studierenden in der Lage sind, IT-gestütztes Projektmanagement und -Governance (K.9, K.11) für eigene IT-Projekte anwenden zu können.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs/Übungsteil und die Studierenden erstellen jeweils individuell ein Konzept für die Planung und Management eines eigenen IT-Projekts.

Wozu: Kompetenzen in der Planung, Organisation und Prozessmanagement von informationstechnischen Systemen und Projekten sind essentiell für technische Informatiker*innen, die im HF3 arbeiten wollen. Durch die Planungsarbeit am eigenen IT-Projekt erwerben die Studierenden zudem Erfahrungen, die essentiell für das HF 1 und 2 sind, u.a. Anforderungen erfassen, Konzepte zu technischen Lösungen entwickeln, das zu entwickelnde Produkt bzw. System analysieren und bewerten. Die Durchführung mit beteiligten Stakeholdern vom Projekt stärkt die Vermittlungs- und Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte

Projekt

Planung der Bachelorarbeit als ein IT-Projekt mit schriftlicher Dokumentation, Präsentation und Verteidigung in einem Gespräch

seminaristischer Unterricht

Projektphasen, Anforderungen und Anforderungsanalyse, Projektplanung, Aufwandsschätzung, Risikomanagement, Netzplantechnik I, Personalführung, Qualitätsmanagement, Fortschrittskontrolle, Projektabschluss, IT-gestütztes Projektmanagement, Projekt-Governance, Projektplanung in wissenschaftlichen Arbeiten

Lehr- und Lernmethoden	▪ Projekt ▪ seminaristischer Unterricht
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	34 Stunden $\pm$ 3 SWS
Selbststudium	116 Stunden
Empfohlene Voraussetzungen	Erfahrung mit Entwurf und Entwicklung von Software

**Zwingende
Voraussetzungen**

Empfohlene Literatur ■ Hindel et. al.: Basiswissen Softwareprojektmanagement, dpunkt.verlag, 2009.

**Verwendung des
Moduls in
weiteren Studiengängen**

**Besonderheiten und
Hinweise**

Letzte Aktualisierung 19.7.2025, 14:32:16

6.30 ITS - IT-Sicherheit

Modulkürzel	ITS_BaTIN2020
Modulbezeichnung	IT-Sicherheit
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	ITS - IT-Sicherheit
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Heiko Knospe/Professor Fakultät IME
Dozierende*r	Prof. Dr. Heiko Knospe/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt die grundlegenden Konzepte und Verfahren der IT-Sicherheit, die für viele IT-Systeme und Anwendungen eine wichtige Rolle spielen (K. 4). Die Studierenden lernen die Analyse von Systemen in Bezug auf Sicherheitsanforderungen (K. 7). Hierfür ist ein Verständnis von Sicherheitsbedrohungen und Angriffen notwendig. Die Studierenden lernen die grundlegenden Verfahren und Standards der IT-Sicherheit um Systeme zu entwerfen, zu realisieren und zu prüfen (K. 8, K. 9, K. 10). Ethische Grundwerte spielen in diesem Zusammenhang eine wichtige Rolle (K. 18), z.B. beim Umgang mit personenbezogenen Daten, Womit: Der Dozent/die Dozentin vermittelt Wissen und Basisfertigkeiten in der Vorlesung. In der Übung bearbeiten die Studierenden unter Anleitung Aufgaben. Im Praktikum werden konkrete Probleme und Fragestellungen der IT-Sicherheit bearbeitet.

Wozu: Grundlegende Kenntnisse der IT-Sicherheit werden in mehreren Modulen des Studiengangs verwendet und sind anerkannter Teil der Basisausbildung in technischen Fächern (HF 1). Bei der Planung von Systemen für technische Anwendungen, der Analyse und Bewertung von Anforderungen sowie der Administration von IT-Systemen spielen Fragen der IT-Sicherheit heute eine wichtige Rolle (HF 5). Die Sicherheit von IT-Systemen ist Teil der Qualitätskontrolle und kann auch in Zertifizierungsprozessen von Bedeutung sein (HF 2).

Modulinhalte**Vorlesung / Übungen**

Grundlagen der IT-Sicherheit: Standards und Richtlinien, Taxonomie, Sicherheitsziele, Bedrohungen, Risiko, Angriffe, Maßnahmen.

Verfahren der Kryptographie: mathematische und algebraische Grundlagen, Definitionen von Sicherheit, historische Chiffren, symmetrische Verschlüsselung, Blockchiffren, Betriebsmodi, Stromchiffren, Hashverfahren, Message Authentication Codes, asymmetrische Verschlüsselung, RSA, Schlüsselvereinbarung, Diffie-Hellman, Signaturverfahren.

Authentisierung, Schlüsselvereinbarung und Zugriffskontrolle: Verfahren der Authentisierung, Passwörter, Schlüsselvereinbarung, Protokolle, öffentliche Schlüssel und Public-Key Infrastrukturen (PKI), Strategien der Zugriffskontrolle, Zugriffsmatrix, Unix ACL.

Netzwerksicherheit: Protokolle TLS und SSH, Sicherheit von DNS.

Software- und Websicherheit: Grundlegende Prinzipien und Design sicherer Software, Schwachstellen, Angriffe gegen Webanwendungen.

Sicherheitsmanagement: Risikomanagement, Organisation des Sicherheitsprozesses, Sicherheitsstandards, insbesondere ISO 27000 Reihe und IT-Grundschutz, Datenschutz (Privacy), Gesetze, ethische Aspekte.

Praktikum

- Erarbeitung von Grundlagen der Cyber-Sicherheit (E-Learning).
- Erstellung von Java Software zur AES Verschlüsselung und Entschlüsselung von Files.
- Einsatz unterschiedlicher Betriebsmodi für Blockchiffren.
- Statistische Analyse eines AES Chiffretextes.
- Erzeugung von Schlüsselpaaren, Zertifikaten und Aufbau einer Public-Key Infrastruktur mit Open Source Software (optional).
- Einsatz eines Linux-Systems für Penetrationstests und digitale Forensik (Kali Linux).
- Angriffe gegen schwache Passwörter.
- Angriffe gegen Webanwendungen (Testsystem).
- Einsatz von Software zur Erkennung und Analyse von Schwachstellen.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen
	▪ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul PI1: -
	▪ Modul PI2: -
	▪ Modul NP: -
	▪ Modul MA1: -
	▪ Modul MA2: -
	▪ - Programmier-Kenntnisse, insbesondere Java, C und Skriptsprachen.
	- Betriebssystem-Kenntnisse, insbesondere Linux.
	- Datennetz-Kenntnisse, insbesondere TCP/IP.
	- Mathematik-Kenntnisse, insbesondere Mengen, Abbildungen, Restklassen, lineare Abbildungen.

Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 3 Termine
	▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum

-
- | | |
|-----------------------------|---|
| Empfohlene Literatur | <ul style="list-style-type: none">▪ C. Eckert, IT-Sicherheit, Oldenbourg Verlag▪ D. Gollmann, Computer Security, John Wiley & Sons▪ J. Schwenk, Sicherheit und Kryptographie im Internet, Springer Verlag▪ G. Schäfer, M. Roßberg, Netzsicherheit, dpunkt Verlag▪ W. Stallings, L. Brown, Computer Security: Principles and Practice, Pearson▪ N. Pohlmann, Cyber-Sicherheit, Springer Verlag▪ H. Knospe, A Course in Cryptography, American Mathematical Society▪ H. Kersten, G. Klett, J. Reuter, K.-W. Schröder, IT-Sicherheitsmanagement nach der neuen ISO 27001. Springer.▪ C. Paar, J. Pelzl, Kryptografie verständlich, Springer.▪ P. C. van Oorschot, Computer Security and the Internet, Springer.▪ C. Pfleeger et al., Security in Computing, Pearson▪ J. Schwenk, Guide to Internet Cryptography, Springer |
|-----------------------------|---|

-
- | | |
|--|---|
| Verwendung des Moduls in weiteren Studiengängen | <ul style="list-style-type: none">▪ ITS in Bachelor Elektrotechnik 2020▪ ITS in Bachelor Elektrotechnik und Informationstechnik 2024▪ ITS in Bachelor Informatik und Systems-Engineering 2024 |
|--|---|
-

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.31 KOAK - Kommunikationsakustik

Modulkürzel	KOAK_BaTIN2020
Modulbezeichnung	Kommunikationsakustik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	KOAK - Kommunikationsakustik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr.-Ing. Christoph Pörschmann/Professor Fakultät IME
Dozierende*r	Prof. Dr.-Ing. Christoph Pörschmann/Professor Fakultät IME

Learning Outcome(s)

Was: Durch das Modul lernen die Studierenden, die grundlegenden Konzepte und physikalischen Zusammenhänge der Akustik auf diverse Anwendungen zu beziehen. Sie werden in die Lage versetzt, diese Anwendungen zu beschreiben, zu analysieren und die Auswirkungen veränderter Randbedingungen abzuschätzen.

Womit: Durch das Anwenden der Grundlagen auf diverse Problemstellungen verstehen die Studierenden viele praktische Anwendungen der Akustik. Ein weiteres Verständnis der grundlegenden Zusammenhänge wird durch das Praktikum bewirkt, in dem die Studierenden selbst einige einfache Anwendungen nutzen, erweitern und einsetzen.

Wozu: Akustische Zusammenhänge spielen im Alltag eines Ingenieurs an vielen Stellen eine wesentliche Rolle, vom Lärmschutz, über Grundprinzipien der Schallausbreitung in Räumen. Für medientechnische Systeme und Medienprodukte spielt die gezielte Anregung und kontrollierte Ausbreitung von Schall eine große Rolle. Die Veranstaltung vermittelt hierzu die Anwendungskenntnisse.

Modulinhalte

Vorlesung / Übungen

- Verfahren zur Raumsimulation und die hierzu erforderlichen Softwaretools
- Menschliches Hörsystem, grundlegende Phänomene der auditiven Wahrnehmung, psychoakustischen Größen
- Räumliche Wahrnehmungsfähigkeiten des Menschen
- Prinzipien der menschlichen Spracherzeugung, gängige Verfahren zur Sprachsignalverarbeitung
- Problemstellungen des Schallschutzes und von Lärmproblemen analysieren und lösen.
- Psychoakustischen Größen zu den physikalischen Größen in Bezug setzen
- Analysieren und Anpassung von räumlichen Beschallungssystemen

Praktikum

- Nachhallzeitmessung
- Raumsimulation nutzen
- Audiometrie (Ruhehörschwelle bestimmen)
- textlich beschriebene Aufgaben in praktische Messungen umsetzen
- funktionstüchtige Messaufbauten erstellen
- fachgerechte Dokumentationen für durchgeführte Messungen anfertigen
- Messergebnisse bewerten und diskutieren

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	Grundkenntnisse Mechanik Kenntnisse Zeit- und Frequenzbereich Komplexe Rechnung Grundkenntnisse Integral- und Differentialrechnung Grundkenntnisse Akustik
Zwingende Voraussetzungen	■ Praktikum erfordert Anwesenheit im Umfang von: 3 Praktikumstermine und 1 Hörversuch ■ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	■ Blauert, J.,(2005) „Communication Acoustics,“ Springer Verlag Heidelberg ■ Weinzierl, Stefan (2008). „Handbuch der Audiotechnik,“ Springer Verlag, Berlin. ■ Blauert, J.,(2021) „Acoustics for Communication,“ Springer Verlag Heidelberg, upcoming ■ Veit, I. (2005). „Technische Akustik“, Kamprath-Reihe, Vogel-Verlag, Würzburg. ■ Cremer, L. (1976). „Vorlesungen über Technische Akustik,“ Springer Verlag, Berlin, Heidelberg. ■ Kuttruff, H. (2004). „Akustik – Eine Einführung,“ S. Hirzel Verlag, Stuttgart.
Verwendung des Moduls in weiteren Studiengängen	■ KOAK in Bachelor Elektrotechnik 2020 ■ KOAK in Bachelor Elektrotechnik und Informationstechnik 2024 ■ KOAK in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.32 KOLL - Kolloquium zur Bachelorarbeit

Modulkürzel	KOLL_BaTIN2020
Modulbezeichnung	Kolloquium zur Bachelorarbeit
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	BAKOLL - Kolloquium
ECTS credits	3
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	7
Häufigkeit des Angebots	jedes Semester
Modul-Verantwortliche*r	Studiengangsleiter(in) Bachelor Technische Informatik / Informatik und Systems-Engineering
Dozierende*r	verschiedene Dozenten*innen / diverse lecturers

Learning Outcome(s)

WAS:

Fachliche und außerfachliche Bezüge der eigenen Arbeit darstellen, bewerten und begründen.

WOMIT:

Präsentationstechniken (schriftlich als auch mündlich) sowie kritische Reflexion der eigenen Arbeitsergebnisse

WOZU:

Um eigene Lösungswege und gewonnene Erkenntnisse vor Fachpublikum darstellen, bewerten und diskutieren zu können.

WAS:

Eigene Arbeitsweise und Ergebnisse präsentieren.

WOMIT:

Präsentationstechniken (schriftlich als auch mündlich) sowie sowie kritische Reflexion der eigenen Arbeitsweise.

WOZU:

Um eigene Lösungswege und gewonnene Erkenntnisse vor Fachpublikum darstellen, bewerten und diskutieren zu können.

Modulinhalte**Kolloquium**

Das Kolloquium dient der Feststellung, ob die Studentin oder der Student befähigt ist, die Ergebnisse der Bachelorarbeit, ihre fachlichen und methodischen Grundlagen, fachübergreifende Zusammenhänge und außerfachlichen Bezüge mündlich darzustellen, selbständig zu begründen und ihre Bedeutung für die Praxis einzuschätzen

Lehr- und Lernmethoden	Kolloquium
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	90 Stunden
Präsenzzeit	0 Stunden $\triangleq$ 0 SWS
Selbststudium	90 Stunden
Empfohlene Voraussetzungen	

Zwingende Voraussetzungen	▪ Modul BAA: Die Bachelorarbeit muss abgeschlossen sein, damit sie im Kolloquium ganzheitlich und abschließend präsentiert werden kann. ▪ Siehe Prüfungsordnung §29, Abs. 2
----------------------------------	---

Empfohlene Literatur

Verwendung des Moduls in weiteren Studiengängen	▪ KOLL in Bachelor Elektrotechnik 2020 ▪ KOLL in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ KOLL in Bachelor Medientechnologie 2020 ▪ KOLL in Bachelor Medientechnologie 2024 ▪ KOLL in Bachelor Optometrie 2021 ▪ KOLL in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise	Siehe auch Prüfungsordnung §29.
------------------------------------	---------------------------------

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.33 MA1 - Mathematik 1

Modulkürzel	MA1_BaTIN2020
Modulbezeichnung	Mathematik 1
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	MA1 - Mathematik 1
ECTS credits	10
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	1
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Heiko Knospe/Professor Fakultät IME
Dozierende*r	▪ Prof. Dr. Heiko Knospe/Professor Fakultät IME ▪ Prof. Dr. Hubert Randerath/Professor Fakultät IME ▪ Prof. Dr. Beate Rhein/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt die grundlegenden Konzepte und Methoden der Mathematik, die in der Informatik und Technik benötigt werden (K. 3). Die Abstraktion und mathematischen Formalisierung von Problemen soll erlernt und angewendet werden (K. 2). Die Studierenden lernen in der Mathematik die Grundzüge wissenschaftlichen Arbeitens kennen (K. 12).

Womit: Der Dozent/die Dozentin vermittelt Wissen und Basisfertigkeiten in der Vorlesung. In der Übung bearbeiten die Studierenden unter Anleitung Aufgaben. Die Übung wird durch Hausaufgaben und Online-Aufgaben (E-Learning) ergänzt. Zusätzlich findet ein Tutorium statt.

Wozu: Grundlegende Mathematik-Kenntnisse werden in mehreren Modulen des Studiengangs benötigt und sind anerkannter Teil der Basisausbildung. Mathematische Methoden sind essentiell für Informatiker, die Systeme zur Verarbeitung, Übertragung und Speicherung von Informationen für technische Anwendungen planen, realisieren und integrieren (HF 1). Die Analyse und Bewertung von Anforderungen, Konzepten und Systemen erfordert häufig mathematische Methoden (HF 2).

Modulinhalte**Vorlesung / Übungen**

Grundlagen

- Mengen, Zahlen, Summen, Produkte, Fakultät, Binomialkoeffizienten
- Reelle Zahlen, Anordnung, Intervalle, Betrag, Vollständigkeit
- Aussagenlogik
- Vollständige Induktion
- Abbildungen und ihre Eigenschaften
- Reelle Funktionen, Beschränktheit, Monotonie, Umkehrfunktion

Elementare Funktionen

- Polynome und rationale Funktionen
- Potenz-, Wurzel-, Exponential-, Logarithmusfunktionen
- Trigonometrische Funktionen

Folgen, Reihen und Stetigkeit

- Reelle Folgen und Grenzwerte
- Reihen und (optional) Konvergenzkriterien
- Potenzreihen und (optional) Konvergenzradius
- Grenzwerte von Funktionswerten
- Stetigkeit und Eigenschaften stetiger Funktionen
- Asymptoten

Differentialrechnung

- Differenzierbarkeit und Ableitung
- Ableitungsregeln
- Höhere Ableitungen
- Extremstellen und Kurvendiskussion
- Taylor-Polynom, Taylor-Reihe
- Newton-Verfahren
- Regel von de l'Hospital

Vektoren, Matrizen und lineare Gleichungssysteme

- Vektorrechnung im $\mathbb{R}^n$
- Skalarprodukt
- Vektorprodukt
- Geraden
- Ebenen
- Matrizen und ihre Rechenregeln
- Lineare Gleichungssysteme und Gaußscher Algorithmus
- Lineare Unabhängigkeit, Erzeugendensystem und Basis
- Rang einer Matrix
- Quadratische Matrizen und invertierbare Matrizen
- Determinante
- Cramersche Regel (optional)

Komplexe Zahlen

- Normalform und Rechenregeln
- Polar- und Exponentialform
- Komplexe Folgen, Reihen, Funktionen, Potenzreihen, Eulersche Formel
- Potenzen und Wurzeln

Übungen / Praktikum

Online Mathematik Kurs OMB+ mit den Inhalten:

- Mengen, Zahlen, Bruchrechnung
- Wurzeln, Potenzen, Proportionalität
- Gleichungen in einer Unbekannten

**Lehr- und
Lernmethoden**

- Vorlesung / Übungen
- Übungen / Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	300 Stunden
Präsenzzeit	57 Stunden $\pm$ 5 SWS
Selbststudium	243 Stunden
Empfohlene Voraussetzungen	Schulkenntnisse Mathematik und Vorkurs oder Brückenkurs Mathematik, insbesondere: Zahlen, Bruchrechnen, Terme, Gleichungen, Funktionen, Geraden, quadratische Funktionen, Polynome, Nullstellen, rationale Funktionen, Wurzel-, Potenz, Exponential- und Logarithmusfunktionen, trigonometrische Funktionen, elementare Geometrie, Vektorrechnung, Geraden, Ebenen, Lösung von linearen Gleichungssystemen (mit zwei oder drei Variablen).
Zwingende Voraussetzungen	Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Übungen / Praktikum
Empfohlene Literatur	▪ P. Hartmann, Mathematik für Informatiker, Vieweg Verlag ▪ T. Westermann, Mathematik für Ingenieure, Springer Verlag ▪ T. Rießinger, Mathematik für Ingenieure, Springer Verlag ▪ M. Knorrenschild, Mathematik für Ingenieure 1, Hanser Verlag ▪ W. Schäfer, G. Trippler, G. Engeln-Müllges (Hrsg.), Kompaktkurs Ingenieurmathematik, Fachbuchverlag Leipzig ▪ L. Papula, Mathematik für Ingenieure und Naturwissenschaftler, Band 1 und 2, Vieweg+Teubner Verlag ▪ G. Hoever, Höhere Mathematik kompakt, Springer Verlag ▪ O. Forster, Analysis 1, Vieweg Verlag ▪ C. Blatter, Analysis 1, Springer Verlag ▪ hm4mint.nrw, Online-Kurs Höhere Mathematik 1 ▪ M. Spivak, Calculus, Cambridge University Press ▪ G. Strang, Lineare Algebra, Springer Verlag ▪ H. Grauert, I. Lieb, Differential- und Integralrechnung I, Springer Verlag ▪ W. Walter, Analysis 1, Springer Verlag
Verwendung des Moduls in weiteren Studiengängen	▪ MA1 in Bachelor Elektrotechnik 2020 ▪ MA1 in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ MA1 in Bachelor Medientechnologie 2020 ▪ MA1 in Bachelor Optometrie 2021 ▪ MA1 in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.34 MA2 - Mathematik 2

Modulkürzel	MA2_BaTIN2020
Modulbezeichnung	Mathematik 2
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	MA2 - Mathematik 2
ECTS credits	10
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	2
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Heiko Knospe/Professor Fakultät IME
Dozierende*r	▪ Prof. Dr. Heiko Knospe/Professor Fakultät IME ▪ Prof. Dr. Hubert Randerath/Professor Fakultät IME ▪ Prof. Dr. Beate Rhein/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul schließt an das Modul "Mathematik 1" an und setzt die Vermittlung grundlegender Konzepte und Methoden der Mathematik fort, die in der Informatik und Technik benötigt werden (K. 3). Die Abstraktion und mathematischen Formalisierung von Problemen soll erlernt und angewendet werden (K. 2). Die Studierenden lernen in der Mathematik die Grundzüge wissenschaftlichen Arbeitens kennen (K. 12).

Womit: Der Dozent/die Dozentin vermittelt Wissen und Basisfertigkeiten in der Vorlesung. In der Übung bearbeiten die Studierenden unter Anleitung Aufgaben. Die Übung wird durch Hausaufgaben und Online-Aufgaben (E-Learning) ergänzt. Zusätzlich findet ein Tutorium statt.

Wozu: Grundlegende Mathematik-Kenntnisse werden in mehreren Modulen des Studiengangs benötigt und sind anerkannter Teil der Basisausbildung. Mathematische Methoden sind essentiell für Informatiker, die Systeme zur Verarbeitung, Übertragung und Speicherung von Informationen für technische Anwendungen planen, realisieren und integrieren (HF 1). Die Analyse und Bewertung von Anforderungen, Konzepten und Systemen erfordert häufig mathematische Methoden (HF 2).

Modulinhalte**Vorlesung / Übungen**

Integralrechnung

- Riemann-Integral, Definition und Eigenschaften
- Hauptsatz der Differential- und Integralrechnung
- Uneigentliche Integrale
- Partielle Integration
- Substitutionsregel
- Partialbruchzerlegung

Gewöhnliche Differentialgleichungen

- DGL erster Ordnung mit trennbaren Variablen
- Lineare DGL erster Ordnung mit konstanten Koeffizienten
- Lineare DGL zweiter Ordnung mit konstanten Koeffizienten

Funktionen von mehreren Variablen

- Skalarfunktionen und Vektorfelder
- Grenzwert und Stetigkeit
- Partielle Ableitungen und Gradient
- Jacobi-Matrix
- Höhere partielle Ableitungen
- Extremwerte
- Fehlerfortpflanzung
- Implizite Funktionen
- Mehrdimensionale Integration

Vektorräume und lineare Abbildungen

- Gruppen, Körper, Endliche Körper
- Vektorräume und Untervektorräume
- Lineare Abbildungen
- Lineare Unabhängigkeit, Dimension und Rang
- Determinante
- Euklidische und unitäre Vektorräume, Skalarprodukt, Norm, Gram-Schmidt Orthogonalisierung
- Orthogonale und unitäre Matrizen
- Symmetrische und Hermitesche Matrizen
- Eigenwerte und Eigenvektoren
- Koordinaten und Basiswechsel
- Diagonalisierbare Matrizen und Normalformen (optional)
- Matrixzerlegungen (optional)
- Homogene Koordinaten (optional)

Übungen / Praktikum

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Übungen / Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	300 Stunden
Präsenzzeit	57 Stunden $\pm$ 5 SWS
Selbststudium	243 Stunden
Empfohlene Voraussetzungen	▪ Modul MA1: Differential- und Integralrechnung von Funktionen einer Variablen sowie Grundlagen der Linearen Algebra. ▪ Mathematik 1, insbesondere: Grundlagen, Elementare Funktionen, Folgen, Reihen, Stetigkeit, Differentialrechnung, Vektoren, Matrizen, lineare Gleichungssysteme.

**Zwingende
Voraussetzungen****Empfohlene Literatur**

- P. Hartmann, Mathematik für Informatiker, Vieweg Verlag
- T. Westermann, Mathematik für Ingenieure, Springer Verlag
- T. Rießinger, Mathematik für Ingenieure, Springer Verlag
- W. Schäfer, G. Trippler, G. Engeln-Müllges (Hrsg.), Kompaktkurs Ingenieurmathematik, Fachbuchverlag Leipzig
- L. Papula, Mathematik für Ingenieure und Naturwissenschaftler, Band 1 und 2, Vieweg+Teubner Verlag
- G. Strang, Lineare Algebra, Springer Verlag
- G. Fischer, Lineare Algebra, Springer Verlag
- D. C. Lay, Linear Algebra and its Applications, Addison Wesley Verlag
- C. Blatter, Analysis 1 und Analysis 2, Springer Verlag
- W. Walter, Analysis 1 und Analysis 2, Springer Verlag
- O. Forster, Analysis 1 und Analysis 2, Springer Verlag
- M. Knorrenschild, Mathematik für Ingenieure 2, Hanser Verlag

**Verwendung des
Moduls in
weiteren Studiengängen**

- MA2 in Bachelor Elektrotechnik 2020
- MA2 in Bachelor Elektrotechnik und Informationstechnik 2024
- MA2 in Bachelor Medientechnologie 2020
- MA2 in Bachelor Informatik und Systems-Engineering 2024

**Besonderheiten und
Hinweise****Letzte Aktualisierung** 19.7.2025, 14:32:16

6.35 ML - Maschinelles Lernen

Modulkürzel	ML_BaTIN2020
Modulbezeichnung	Maschinelles Lernen
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	ML - Maschinelles Lernen
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Beate Rhein/Professor Fakultät IME
Dozierende*r	Prof. Dr. Beate Rhein/Professor Fakultät IME

Learning Outcome(s)

Was: Die Studierenden erlernen die Methoden und Verfahren zum Maschinellen Lernen mit Neuronalen Netzen. Darauf aufbauend wird die Problemlösung mittels Entwurfswerkzeugen implementiert und validiert.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. Im Praktikum erarbeiten die Studierenden in Kleingruppen Problemlösungen und verteidigen diese.

Wozu: Die Studierenden werden in dem Modul befähigt, im Problemlösung im Bereich des maschinellen Lernens mit neuronalen Netzen zu lösen und bestehende Lösungen zu bewerten.

Modulinhalte**Vorlesung / Übungen**

Allgemeine Grundlagen
 Performanzmaße
 Gradientenabstiegsverfahren
 Verfahren zur Dimensionsreduzierung
 Einfache neuronale Netze
 Backpropagation-Trainingsalgorithmus
 Deep Neural Networks
 Typische Probleme von Deep Neural Networks und Lösungsansätze hierzu
 Convolutional Neural Networks (CNNs)
 Rekurrente Neuronale Netze (RNN)
 Die vorgestellten neuronalen Netze
 angeben
 beschreiben
 hinsichtlich der Einsatzfelder abgrenzen
 hinsichtlich der Vor- und Nachteile bewerten
 Lösung von Problemstellungen unter Verwendung von Tools
 zur Handhabung, Analyse und Aufbereitung der Trainingsdaten
 zur Erstellung, Verifikation, Validierung und zum Training aller vorgestellten Neuralen Netzen

Praktikum

siehe Fertigkeiten, die unter "Vorlesung/Übung->Lernziele->Fertigkeiten" aufgeführt sind
 komplexere Aufgaben in einem Kleinteam bewältigen
 Erarbeitung von komplexeren Problemlösungen die sich mit neuronalen Netzen implementieren lassen
 komplexere Problemstellungen verstehen und analysieren
 Systemverhalten aus spezifizierenden Texten herleiten
 System strukturiert analysieren
 sinnvolle Teilsysteme erkennen
 Schnittstellen zwischen Teilsystemen erfassen
 Gesamtsystem auf Basis von Teilsystemen modellieren
 Auswahl geeigneter bekannter Verfahren/Netze
 Modifikation bekannter Verfahren
 Kombination geeigneter Verfahren

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul MA1: Problemlösungskompetenz aus dem Bereich lineare Algebra und der Analysis. Sicheres Beherrschen der entsprechenden Symbole und Formalismen. ▪ Grundlagen der Programmierung in Java oder C ▪ Grundlagen der Analysis und Linearen Algebra
Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 4 Testattermine ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum

Empfohlene Literatur	▪ Géron, Aurélien, Hands-On Machine Learning with Scikit-Learn and TensorFlow, O'Reilly Medi
-----------------------------	--

Verwendung des Moduls in weiteren Studiengängen	ML in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise	
------------------------------------	--

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.36 MLO - Machine Learning Operations

Modulkürzel	MLO_BaTIN2020
Modulbezeichnung	Machine Learning Operations
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	MLO - Machine Learning Operations
ECTS credits	5
Sprache	deutsch und englisch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Erstmaliges Angebot	Wintersemester 2025
Modul-Verantwortliche*r	Prof. Dr. Pascal Cerfontaine/Professor Fakultät IME
Dozierende*r	Prof. Dr. Pascal Cerfontaine/Professor Fakultät IME

Learning Outcome(s)

(WAS) Die Studierenden erwerben Kenntnisse und Fertigkeiten für die Entwicklung, Bereitstellung, Verwaltung, Überwachung und Wartung von Machine-Learning-Systemen im produktiven Einsatz. Sie verstehen den Lebenszyklus von ML-Modellen und lernen, diesen systematisch zu gestalten und zu automatisieren,
(WOMIT) indem sie in Vorlesungen/Übungen Konzepte, Werkzeuge und Best Practices der MLOps kennenlernen und diese in aufeinander aufbauenden praktischen Aufgaben selbst anwenden,
(WOZU) um robuste, skalierbare und reproduzierbare ML-Anwendungen zu entwickeln, die auch unter realen Bedingungen zuverlässig funktionieren und in Produktionsumgebungen betrieben werden können.

Modulinhalte

Vorlesung / Übungen

- Einführung in Methoden des maschinellen Lernens zur Erstellung von KI-Anwendungen
- Einführung in den ML-Lebenszyklus: Entwicklung, Datenvorbereitung, Modelltraining und -überprüfung, Bereitstellung, Monitoring, Retraining
- Unterschiede zwischen klassischem Software-Engineering und ML-Systementwicklung
- Grundlagen reproduzierbarer ML-Experimente und des Daten- und Modellmanagements
- Versionsverwaltung für Modelle und Datensätze (z. B. DVC, MLflow, Git)
- Containerisierung von ML-Modellen mit Docker
- Modell-Serving (z. B. REST-APIs mit FastAPI, TensorFlow Serving, TorchServe)
- Automatisiertes Training und Bereitstellung
- Monitoring von ML-Systemen: Modell-Drift, Daten-Drift, Performance-Metriken, Logging, Alerting
- Orchestrierung von ML-Workflows: Verwendung von Tools wie Kubeflow, Airflow oder Prefect
- Skalierung mit Cloud-Diensten (z. B. Vertex AI, AWS Sagemaker, Azure ML)
- Sicherheit, Datenschutz und ethische Aspekte beim ML-Betrieb

Praktikum

- Aufbereitung und Versionierung von Daten für ML-Workflows
- Erstellen einer KI-Anwendungen inklusive Modelltraining
- Reproduzierbares Modelltraining mit DVC oder MLflow
- Containerisierung und Bereitstellung eines ML-Modells als Webservice
- Entwicklung einer CI/CD-Pipeline für ein ML-System (z. B. mit GitHub Actions oder GitLab CI)
- Einrichtung eines Monitoringsystems (Prometheus, Evidently, Grafana) zur Überwachung von Modellen und Datenströmen
- Durchführung und Bewertung von Retraining-Zyklen basierend auf realistischen Szenarien
- Integration und Deployment in einer Cloud-Umgebung

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\triangleq$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul DOPE: Grundlagen des Versionsverwaltungssystems Git werden beherrscht ▪ Modul DOPE: Containerisierung mit Docker ist bekannt ▪ Modul ML: Grundlagen des maschinellen Lernens sind bekannt ▪ - Grundlagen des Versionsverwaltungssystems Git werden beherrscht - Containerisierung mit Docker ist bekannt - Grundlegende Kenntnisse in Python
Zwingende Voraussetzungen	
Empfohlene Literatur	▪ Yaron Haviv, Noah Gift: Implementing MLOps in the Enterprise, 2023 ▪ How to Learn MLOps in 2024: Courses, Books, and Other Resources
Verwendung des Moduls in weiteren Studiengängen	MLO in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.8.2025, 10:30:37

6.37 MT - Messtechnik

Modulkürzel	MT_BaTIN2020
Modulbezeichnung	Messtechnik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	MT - Messtechnik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Kai Kreisköther/Professor Fakultät IME
Dozierende*r	Prof. Dr. Kai Kreisköther/Professor Fakultät IME
Learning Outcome(s)	
Messabweichungen kennenlernen und analysieren	
Kennenlernen und Anwenden der Grundlagen der Stochastik	
Kennenlernen und Analysieren statistischer Größen	
Analoge Messgeräte kennenlernen und anwenden	
Digitale Messgeräte kennenlernen und anwenden	
Messverfahren und Sensorik verstehen und anwenden	

Modulinhalte**Vorlesung / Übungen**

Einführung in die Messtechnik

- Allgemeine Betrachtungen
- Historischer Rückblick
- Das SI-System
- Begriffsdefinitionen
- Messtechnische Tätigkeiten
- Messmethoden

Messabweichungen

- Bekannte systematische Messabweichungen
- Unbekannte systematische Messabweichungen
- Fortpflanzung systematischer Messabweichungen
- Zufällige Messabweichungen
- Fortpflanzung zufälliger Messabweichungen
- Messunsicherheit und vollständiges Messergebnis

Analoge Messgeräte und Messverfahren

- Eigenschaften elektrischer Messgeräte
- Elektro-mechanische Messwerke

Gleichstrom- und Gleichspannungsmessung

- Grundsaltungen
- Messbereichserweiterungen
- Begrenzerschaltungen mit Dioden

Wechselstrom- und Wechselspannungsmessung

- Beschreibung periodisch zeitabhängiger Größen
- Messgleichrichter

Digitale Messgeräte

- A/D- und D/A-Wandlung
- Digitale Multimeter
- Digitale Oszilloskope
- Zeit- und Frequenzmessung

Messanwendungen und Sensorik – Widerstands-Messbrücken – Wechselspannungs-Messbrücken – Sensoren

Praktikum

Umgang mit digitalen Oszilloskopen verstehen und anwenden

Analysieren von Begrenzer-Schaltung

Analyse von galvanischen, magnetischen und kapazitiven Kopplungen

Lehr- und Lernmethoden	▪ Vorlesung / Übungen
	▪ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul MA1: Gleichungssysteme lösen ▪ Modul MA2: Differentialrechnung / Integralrechnung ▪ MA1, MA2, GE1, GE2
Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: 3 Praktikumsversuche
Empfohlene Literatur	▪ Schröder, E.: Elektrische Messtechnik ▪ Lerch, R.: Kaltenbacher, M.; Lindinger, F.: Übungen zur Elektrischen Messtechnik ▪ Felderhoff, R.: Elektrische und elektronische Messtechnik ▪ Weichert, N.: Messtechnik und Messdatenerfassung
Verwendung des Moduls in weiteren Studiengängen	▪ MT in Bachelor Elektrotechnik 2020 ▪ MT in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ MT in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.38 NDQ - Nachhaltigkeit durch Qualität

Modulkürzel	NDQ_BaTIN2020
Modulbezeichnung	Nachhaltigkeit durch Qualität
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	NDQ - Nachhaltigkeit durch Qualität
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Ansgar Beuten/Lehrbeauftragter
Dozierende*r	Ansgar Beuten/Lehrbeauftragter

Learning Outcome(s)

- Die Studierenden kennen die verschiedenen Formen von Nachhaltigkeit (ökologisch, ökonomisch, sozial), können diese voneinander abgrenzen und im Kontext erläutern.
- Die Studierenden können für die verschiedenen Formen von Nachhaltigkeit Ziele definieren, Kennzahlen ableiten und Ansätze im Hinblick auf Nachhaltigkeit bewerten.
- Die Studierenden können Nachhaltigkeit zielgruppenspezifisch argumentieren und fachlich vertreten.
- Die Studierenden sind in der Lage das Mindset eines Gegenübers in Themen der Nachhaltigkeit positiv zu verändern.
- Die Studierenden können verschiedene Arten von Qualität benennen, erkennen, erklären und differenzieren.
- Die Studierenden können verschiedene Methoden des Qualitätsmanagements erkennen, erklären, differenzieren und anwenden.
- Die Studierenden kennen verschiedene Werkzeuge des Qualitätsmanagements und können diese erklären und anwenden.
- Die Studierenden sind in der Lage, Verbindung zwischen Nachhaltigkeit und Qualität herzustellen, Abhängigkeiten zu erkennen und zu analysieren. Die Studierenden können durch Anwenden der erlernten Methoden und Werkzeuge Nachhaltigkeit erzeugen und optimieren.

Modulinhalte

Vorlesung

- Die Studierenden kennen die verschiedenen Formen von Nachhaltigkeit (ökologisch, ökonomisch, sozial), können diese voneinander abgrenzen und im Kontext erläutern.
- Die Studierenden können für die verschiedenen Formen von Nachhaltigkeit Ziele definieren, Kennzahlen ableiten und Ansätze im Hinblick auf Nachhaltigkeit bewerten.
- Die Studierenden können Nachhaltigkeit zielgruppenspezifisch argumentieren und fachlich vertreten.
- Die Studierenden sind in der Lage das Mindset eines Gegenübers in Themen der Nachhaltigkeit positiv zu verändern.
- Die Studierenden können verschiedene Arten von Qualität benennen, erkennen, erklären und differenzieren.
- Die Studierenden können verschiedene Methoden des Qualitätsmanagements erkennen, erklären, differenzieren und anwenden.
- Die Studierenden kennen verschiedene Werkzeuge des Qualitätsmanagements und können diese erklären und anwenden.
- Die Studierenden sind in der Lage, Verbindung zwischen Nachhaltigkeit und Qualität herzustellen, Abhängigkeiten zu erkennen und zu analysieren. Die Studierenden können durch Anwenden der erlernten Methoden und Werkzeuge Nachhaltigkeit erzeugen und optimieren.

seminaristischer Unterricht

- identisch zu Vorlesung

Lehr- und Lernmethoden	▪ Vorlesung ▪ seminaristischer Unterricht
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul MA1: erforderlich für das Verständnis statistischer Methoden ▪ Modul MA2: erforderlich für das Verständnis statistischer Methoden ▪ Mathematik 1 und Mathematik 2, um bei den Werkzeugen des Qualitätsmanagements ein Verständnis für die statistischen Methoden zu ermöglichen.
Zwingende Voraussetzungen	▪ seminaristischer Unterricht erfordert Anwesenheit im Umfang von: An mindesten acht Terminen des Seminars müssen sich die Studierenden anwesend sein und sich beteiligen. ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an seminaristischer Unterricht
Empfohlene Literatur	
Verwendung des Moduls in weiteren Studiengängen	▪ NDQ in Bachelor Elektrotechnik 2020 ▪ NDQ in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ NDQ in Bachelor Medientechnologie 2020 ▪ NDQ in Bachelor Medientechnologie 2024 ▪ NDQ in Bachelor Optometrie 2021 ▪ NDQ in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.39 NP - Netze und Protokolle

Modulkürzel	NP_BaTIN2020
Modulbezeichnung	Netze und Protokolle
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	NP - Netze und Protokolle
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	3
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Andreas Grebe/Professor Fakultät IME
Dozierende*r	Prof. Dr. Andreas Grebe/Professor Fakultät IME

Learning Outcome(s)

Was:

Das Modul vermittelt Wissen zu Kommunikationsprotokollen und deren Rolle und Mechanismen, Wissen zur Architektur und zum Aufbau von Computernetzen sowie ein tieferes Verständnis der zugrundeliegenden Konzepte und Techniken.

Das Modul vermittelt Kompetenzen zur Planung, Implementierung, Evaluierung und zum Betrieb von Computernetzen. Folgende Kenntnisse und Kompetenzen werden im Detail vermittelt: Grundlegende Konzepte und Technologien von Rechnernetzen benennen, strukturieren, einordnen (K.2, K.4, K.8), Strukturieren der Aufgaben in der technischen Kommunikation, zuordnen auf einschlägige Standardisierungen und übertragen auf Netzdesign und Client-/Server-Anwendungen (K.1, K.2, K.3, K.8), Protokolle (Anwendungen, Transport, Netzwerk, Ethernet, Übertragungstechnik) zuordnen und benennen, Protokoll-Mechanismen erläutern, Aufgaben und technische Parameter darlegen und strukturieren (K.1, K.2, K.8), Netze und Systeme unter Einsatz geeigneter Tools analysieren und grafisch darstellen (K.4, K.7, K.8, K.9), Systeme in Netze einbinden, Systemkonfiguration planen (K.4, K.5, K.6, K.7, K.10), Netze planen und einrichten (K.4, K.5, K.6, K.7, K.10), Leistungsfähigkeit von Rechnernetzen abschätzen und analysieren (K.2, K.3, K.7, K.8), Information aus englischen Originalquellen und Standards ableiten (K.2, K.8, K.3, K.4, K.15).

Womit:

Kenntnisse und Basisfertigkeiten werden in Vorlesung und Übung vermittelt. Darauf aufbauend werden im Praktikum Kompetenzen und Fertigkeiten ausgebaut und inhaltliche Themen vertieft. Im Praktikum arbeiten die Studierenden in Kleingruppen und verteidigen ihre Lösungen (K.8, K.16).

Wozu:

Computernetze sind heute die Grundlage für alle technischen Kommunikationssysteme, von der Telekommunikation über Unternehmensnetze bis hin zu Automatisierung und grundlegender Digitalisierung. Sie bilden die Kommunikationsplattform für verteilte Systeme. Entsprechende Kompetenzen und Wissen über die zugehörigen Grundlagen sind essentiell für die Erstellung (HF1), Bewertung (HF2) und Betrieb (HF3) moderner verteilter Systeme und Services. Die Verteidigung der eigenen Lösungen in der Übung und im Praktikum fördert die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte**Vorlesung / Übungen**

Grundlagen von Architekturen und Topologien von Rechnernetzen, Metriken, LAN, MAN WAN, Kommunikations- und Schichtenmodelle nach ISO/OSI, IETF TCP/IP, IEEE, Bitübertragung und Datenverbindungen, Ethernet-Technologie, IP-Adressierung und Subnetting, IP Routing und Routing-Protokolle, Frame-Switching und Virtuelle LAN, Transportprotokolle, Anwendungsprotokolle und Kommunikationsmuster

Netze und Systeme unter Einsatz geeigneter Tools analysieren und grafisch darstellen. Systeme in Netze einbinden. (Sub-)Netze planen und einrichten. Leistungsfähigkeit von Rechnernetzen abschätzen und analysieren. Informationsbeschaffung aus englischen Originalquellen.

Auszug der Inhalte:

ISO/OSI Referenzmodelle, TCP/IP Modell, IEEE Modell, Switch, Router, Host, Übertragungsmedien, Ethernet, 100BASE-Tx, 1000BASE-T, ARP, Adressierung IPv4, IPv6, DHCP, ICMP, Switched LAN, Virtuelle LAN (VLAN), Statisches Routing, RIP, OSPF, Transportprotokolle UDP, TCP, QUIC, Anwendungen DNS, HTTP, FTP, TFTP, Telnet, SSH

Praktikum

Grundlegende Konzepte und Technologien von Rechnernetzen benennen, strukturieren, einordnen, Strukturieren der Aufgaben in der technischen Kommunikation, zuordnen auf einschlägige Standardisierungen und übertragen auf Netzdesign und Client-/Server-Anwendungen, Protokolle (Anwendungen, Transport, Netzwerk, Ethernet, Übertragungstechnik) zuordnen und benennen, Protokoll-Mechanismen erläutern, Aufgaben und technische Parameter darlegen und strukturieren. Netzanalysetechniken und Tools beherrschen, Netzdesignschritte kennen und Methoden zur Netzplanung kennen.

Netze und Systeme unter Einsatz geeigneter Tools analysieren und grafisch darstellen.

Systeme in Netze einbinden.

(Sub-)Netze planen und einrichten.

Leistungsfähigkeit von Rechnernetzen abschätzen und analysieren.

Systematische Fehlersuche und -korrektur vornehmen.

Information aus englischen Originalquellen auswerten in Netzen anwenden.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen
	▪ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul PI1: Sicherer Umgang mit konsolenbasierter Systemsteuerung und einer Programmiersprache inklusive Boole'scher Operationen.
	▪ Modul PI2: Sicherer Umgang mit konsolenbasierter Systemsteuerung und einer Programmiersprache inklusive Boole'scher Operationen.
	▪ Modul DR: Kenntnisse über Aufbau und Funktionalität eines Digitalrechners/Computers.
	▪ Boole'sche Operationen, AND, OR, XOR Binäre Zahlensysteme Rechnerarchitektur (Grundlagen) Grundlegende Kenntnisse eines Betriebssystems (Unix/Linux favorisiert) Grundlegende Kenntnisse strukturierter Programmierung

Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 4 Praktikumsversuche
	▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum

-
- Empfohlene Literatur**
- J. Kurose, K. Ross: Computernetzwerke - Der Top-Down-Ansatz, Pearson Studium, 6. Auflage, 2014
 - A. Tanenbaum: Computernetzwerke, Pearson Studium, 5. Auflage 2012
 - Douglas Comer: Computer Networks and Internets, Pearson Education Limited, 6 edition, 2015
 - Internet-Standardisierung: IETF Standards (RFCs), www.ietf.org
 - LAN-Standards: IEEE, ieeexplore.ieee.org (freier Zugang über TH Köln)
 - Telekommunikationsstandards: ITU-T Standards, www.itu.int
 - Web-Standardisierung: W3C Standards, www.w3c.org

-
- Verwendung des Moduls in weiteren Studiengängen**
- NP in Bachelor Elektrotechnik 2020
 - NP in Bachelor Elektrotechnik und Informationstechnik 2024
 - NP in Bachelor Informatik und Systems-Engineering 2024

Besonderheiten und Hinweise

Letzte Aktualisierung 19.7.2025, 14:32:16

6.40 NSA - Netzsicherheit und Automation

Modulkürzel	NSA_BaTIN2020
Modulbezeichnung	Netzsicherheit und Automation
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	NSA - Netzsicherheit und Automation
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Andreas Grebe/Professor Fakultät IME
Dozierende*r	Prof. Dr. Andreas Grebe/Professor Fakultät IME

Learning Outcome(s)

Das Modul vertieft Wissen und Kompetenzen zu IP-Netzen und Kommunikationsprotokollen. Das Modul vermittelt Kompetenzen zur Planung, Implementierung, Evaluierung und zum Betrieb von größeren, standortübergreifenden Computernetzen inklusive der dazugehörigen Netzsicherheitstechniken und verteilter Netzmanagementtechniken. Zu den Kenntnissen und Kompetenzen gehören: Grundlegende Konzepte und Technologien von skalierenden Rechnernetzen benennen, strukturieren, einordnen (K.2, K.4, K.8), Skalierende Netze unter Einsatz geeigneter Tools analysieren und grafisch darstellen (K.4, K.7, K.8, K.9), planen und einrichten (K.4, K.5, K.6, K.7, K.10), Leistungsfähigkeit von Rechnernetzen abschätzen und analysieren (K.2, K.3, K.7, K.8), Sicherheitsrisiken und Abwehrtechniken erläutern, implementieren und bewerten (K.1, K.2, K.3, K.7, K.8), Netzmanagementaufgaben und -techniken erläutern, implementieren und bewerten (K.1, K.2, K.3, K.7, K.8), Information aus englischen Originalquellen und Standards ableiten (K.2, K.8, K.3, K.4, K.15).

Womit:

Kenntnisse und Basisfertigkeiten werden in Vorlesung und Übung vermittelt. Darauf aufbauend werden im Praktikum Kompetenzen und Fertigkeiten ausgebaut und inhaltliche Themen vertieft. Im Praktikum arbeiten die Studierenden in Kleingruppen und verteidigen ihre Lösungen (K.8, K.16).

Wozu:

Computernetze sind heute die Grundlage für alle technischen Kommunikationssysteme, von der Telekommunikation über Unternehmensnetze bis hin zu Automatisierung und grundlegender Digitalisierung. Das auf dem Modul NP aufbauende Modul IN fokussiert auf Kompetenzen zur Planung, Implementierung (HF1), Betrieb (HF3) und Evaluierung (HF2) von größeren, standortübergreifenden Unternehmensnetzen. Insbesondere durch die Verbindung zum Internet und die standortübergreifenden Aspekte werden Netzsicherheit und Netzmanagement als weitere zusätzliche Schwerpunkt aufgenommen. Die Verteidigung der eigenen Lösungen in der Übung und im Praktikum fördert die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte**Vorlesung / Übungen**

Grundlagen zum Aufbau von hierarchisch strukturierten Netzen, Unternehmensnetzen mit Redundanztechniken, Wireless LAN (WLAN), standortübergreifende Kommunikation, WAN-Techniken. Einführung in die Netzsicherheit mit Vertiefungen zu Angriffen, Sicherheitszielen, kryptographischen Verfahren, Verschlüsselung, Paketfilter, sichere Infrastrukturen, virtuelle private Netze. Einführung in verteiltes Netzmanagement und Servicequalitätstechniken. Techniken zur Netzvirtualisierung, Software-defined Networking und Netzautomatisierung.

Studierende erhalten die Kompetenzen, mittelgroße, standortübergreifende Unternehmensnetze unter Einsatz geeigneter Tools analysieren, geeignete Architekturen auszuwählen und entsprechende Netze zu planen und zu implementieren. Sie benennen und identifizieren Gefährdungslagen für Unternehmensnetze. Geeignete Sicherheitsmechanismen sind auszuwählen, zu designen und zu implementieren. Aufgaben und Methoden softwaregesteuerter Netze inklusive und Virtualisierungen werden benannt und Mechanismen zur Netzautomatisierung geplant und umgesetzt.

Auszug der Inhalte:

Hierarchische Netze, Redundanz, STP, EtherChannel, FHRP, Single-area und Multiarea OSPF, OSPF Sicherheitstechniken, WLAN, WAN-Anschluss, PPP, xDSL

Netzsicherheit mit Sicherheitszielen, kryptographische Verfahren, Algorithmen, Paketfilter, ACL, NAT, Firewall, DMZ, VPN, IPsec, SNMP, Syslog, QoS – Quality-of-Service

Software Defined Networking (SDN), SDN Controller, Cloud, Virtualisierung, Ansible, JSON, YAML, REST API

Praktikum

Konzepte und Technologien für mittelgroße, standortübergreifende Unternehmensnetze benennen, strukturieren, einordnen. Netzanalysetechniken und Tools beherrschen, Netzdesignschritte kennen und Methoden zur Netzplanung kennen. Sicherheitsrelevante Netzaspekte identifizieren und geeignete Massnahmen zur Netzsicherheit und deren Umsetzung kennen. Aufgaben der Netzautomatisierung und Virtualisierung kennen und für geeignete Netzbereiche deren Umsetzung beherrschen.

Planung, Implementierung und Analyse von VLAN-Architekturen, WLAN-Netzen, standortübergreifende VPN und Paketfilter-Firewall. Implementierung und Analyse von Netzmanagement mit SNMP und Syslog.

Implementierung und Analyse von Netzautomatisierung an Netzelementen (u.a. Router, Switch, Host, SDN-Controller) über REST API mit Python-Scripting oder Ansible YAML Skripting.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen
	▪ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul NP: Fundamentale Kenntnisse über IP-basierte Netze und Kompetenzen zu Planung, Implementierung und Betrieb von IP-Netzen werden vorausgesetzt. Im Praktikum werden Kenntnisse zu Protokollanalyse (Wireshark) und Konfiguration / Programmierung von Router und Switch (Cisco IOS-Befehlssätze) erwartet. Die Voraussetzungen können u.a. durch das Testat für die ULP NP oder die nachgewiesenen CCNA Module ITN und RSE nachgewiesen werden.
	▪ Kenntnisse und Kompetenzen des Moduls "Netze und Protokolle (NP)" alternativ: Kenntnisse und Anwendung von grundlegenden Internetworking Techniken Grundlegende Vernetzungstechniken TCP/IP Protokollfamilie ISO/OSI Schichtenmodellierung IPv4/IPv6 Routing Switchingtechniken TCP/UDP Transporttechniken Anwendungsprotokolle Umgang mit Netzelementen (Client, Server, Switch, Router)

Zwingende Voraussetzungen	■ Praktikum erfordert Anwesenheit im Umfang von: 4 Termine ■ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	■ J. Kurose, K. Ross: Computernetzwerke - Der Top-Down-Ansatz, Pearson Studium, 6. Auflage, 2014 ■ A. Tanenbaum: Computernetzwerke, Pearson Studium, 5. Auflage 2012 ■ G. Schäfer: Netzsicherheit: - Grundlagen & Protokolle - Mobile & drahtlose Kommunikation - Schutz von Kommunikationsinfrastrukturen, dpunkt.verlag, 2. Auflage 2014 ■ W. Stallings: Foundations of Modern Networking, Pearson Education, 2016 ■ J. Doherty: SDN and NFV Simplified, Pearson Education, 2016 ■ J. Edelman: Network Programmability and Automation, O'Reilly 2018 ■ Internet-Standardisierung: IETF Standards (RFCs), www.ietf.org ■ LAN-Standards: IEEE, ieeexplore.ieee.org (freier Zugang über TH Köln) ■ Telekommunikationsstandards: ITU-T Standards, www.itu.int ■ Web-Standardisierung: W3C Standards, www.w3c.org
Verwendung des Moduls in weiteren Studiengängen	■ NSA in Bachelor Elektrotechnik 2020 ■ NSA in Bachelor Elektrotechnik und Informationstechnik 2024 ■ NSA in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.41 PHTB - Philosophische Handlungstheorie Bachelor

Modulkürzel	PHTB_BaTIN2020
Modulbezeichnung	Philosophische Handlungstheorie Bachelor
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	PHTB - Philosophische Handlungstheorie Bachelor
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Gregor Büchel/Professor Fakultät IME im Ruhestand
Dozierende*r	Prof. Dr. Gregor Büchel/Professor Fakultät IME im Ruhestand

Learning Outcome(s)

- (WAS) Studierende wenden philosophischen Theorien auf Probleme des Handelns in der heutigen Gesellschaft an,
- (WOMIT) indem Sie zentrale philosophische Texte studieren, seminaristisch aufarbeiten und präsentieren,
- (WOZU) um ihr späteres gesellschaftliches und berufliches Handeln auf philosophisch und ethisch durchdachten Grundlagen aufbauen zu können.

Modulinhalte

Vorlesung

In der Vorlesung wird der Hintergrund philosophischer Handlungstheorien „beleuchtet“

Seminar

Im Zentrum des Seminars steht die Besprechung der fünf folgenden Texte von Immanuel Kant:

1. „Beantwortung der Frage: Was ist Aufklärung?“,
2. „Ideen zu einer allgemeinen Geschichte in weltbürgerlicher Absicht“,
3. „Grundlegung zur Metaphysik der Sitten“,
4. Die Antinomie von Freiheit und Notwendigkeit in der „Kritik der reinen Vernunft“,
5. „Zum ewigen Frieden“.

Aspekte der philosophischen Handlungstheorie, die mit diesen Texten gegeben sind, sollen auf Probleme des Handelns in der heutigen Gesellschaft angewendet werden.

Lehr- und Lernmethoden	▪ Vorlesung ▪ Seminar
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	34 Stunden $\pm$ 3 SWS
Selbststudium	116 Stunden
Empfohlene Voraussetzungen	

Zwingende Voraussetzungen	■ Seminar erfordert Anwesenheit im Umfang von: 6 Termine ■ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Seminar
Empfohlene Literatur	■ Immanuel Kant: „Beantwortung der Frage: Was ist Aufklärung? Und andere kleine Schriften“, Berlin (Sammlung Hoffenberg), 2016, ISBN: 978-3-8430-9208-1 ■ Immanuel Kant: „Schriften zur Geschichtsphilosophie“, Stuttgart (Reclam), 2013, ISBN: 978-3-15-009694-9 ■ Immanuel Kant: „Grundlegung zur Metaphysik der Sitten“, Stuttgart (Reclam), 2016, ISBN: 978-3-15-004507-7 ■ Immanuel Kant: „Kritik der reinen Vernunft“, Stuttgart (Reclam), 1966, ISBN: 978-3-15-006461-0 ■ Immanuel Kant: „Zum ewigen Frieden“, Stuttgart (Reclam), 2012, ISBN: 978-3-15-001501-8
Verwendung des Moduls in weiteren Studiengängen	■ PHTB in Bachelor Elektrotechnik 2020 ■ PHTB in Bachelor Elektrotechnik und Informationstechnik 2024 ■ PHTB in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.42 PI1 - Praktische Informatik 1

Modulkürzel	PI1_BaTIN2020
Modulbezeichnung	Praktische Informatik 1
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	PI1 - Praktische Informatik 1
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	1
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Markus Cremer/Professor Fakultät IME
Dozierende*r	Prof. Dr. Markus Cremer/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt grundlegende Kompetenzen zur Nutzung von Programmiersprachen und entsprechender abstrakterer Darstellungsformen bei der algorithmischen und objektorientierten Lösung von Anwendungsproblemen. Die Studierenden sollen in die Lage versetzt werden, einschlägige Begrifflichkeiten und Techniken im praktischen Umfeld sicher anzuwenden: Aufbauend auf den in der Vorlesung vermittelten Kenntnissen (K3) analysieren die Studierenden Problemstellungen (K2), entwerfen Lösungswege dazu (K5), implementieren sie mit Hilfe von Standardwerkzeugen (K6, K9) und prüfen sie (K7).

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. In den Übungen und insbesondere im Praktikum bearbeiten die Studierenden in Kleingruppen Programmieraufgaben und verteidigen ihre Lösungen (K8, K13, K16).

Wozu: Kompetenzen in der Anwendung von Programmiersprachen sind essentiell für Informatiker/-innen, insbesondere in Hinblick auf die Realisierung informationstechnischer Systeme (HF1). Durch ihre praktische Programmierarbeit erwerben die Studierenden zudem Erfahrungen, die wichtig sind für die Erfassung von Anforderungen, die Entwicklung von Konzepten zur technischen Lösung und zu ihrer Bewertung (HF2). Die Durchführung im Team mit dem Dozenten als "Auftraggeber" stärkt die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte

Vorlesung / Übungen

Informatik und Computer

- Was ist Informatik?
- Algorithmen
- Aufbau eines programmierbaren Digitalrechners

Programmierung – eine allgemeine Einführung

- Höhere Programmiersprachen (prozedural vs. objektorientiert)
- Programmausführung
- Programmdokumentation

Zahlenverarbeitung

- Der Variablenbegriff
- Zahlentypen und -konstanten
- Arithmetische Operatoren
- Zuweisungen
- Ein-/Ausgabe

Aussagenlogische Operationen

Textzeichen

- Standards zur Zeichencodierung
- Zeichen in Java
- Zeichenketten

Kontrollstrukturen

- Grafische Darstellungsformen
- Blöcke
- Verzweigungen
- Schleifen

Methoden

- Grundlegende Prinzipien
- Definition
- Aufruf
- Überladen
- Speicherklassen

Arrays

- Grundlegende Eigenschaften
- Arrays in Java
- Mehrdimensionale Arrays
- Arrays und Methoden
- char-Arrays und Strings

Objektorientierte Programmierung: Objekte und Klassen

- Motivation und Grundlagen
- Objekte mit Attributen
- Objekte mit Attributen und Methoden
- Klassenattribute und -methoden
- Hauptprogramm-Klasse und weitere Klassen

Objektorientierte Programmierung: Klassenstrukturen

- Klassenhierarchien
- Abstrakte Klassen und Interfaces
- Pakete

Praktikum

Umgang mit einer Softwareentwicklungsumgebung und mit git/gitlab. Fehlersuche und -beseitigung in Programmen.

Algorithmen erstellen und umsetzen. Kontrollstrukturen kennen und benutzen können.

Programmieren mit strukturierten Datentypen, insbes. Arrays

 Programmierung mit Methoden

 Programmierung elementarer Operationen auf einfachen Datentypen.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\triangleq$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	keine
Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 3 Termine ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	▪ Ernst, H., Schmidt, J., & Beneken, G. (2023). Grundkurs Informatik: Grundlagen und Konzepte für die erfolgreiche IT-Praxis – Eine umfassende Einführung. Springer Vieweg. https://doi.org/10.1007/978-3-658-41779-6 ▪ Gumm, H.-P., & Sommer, M. (2016–2019). Einführung in die Informatik (3 Bände). De Gruyter / Oldenbourg. ▪ Herold, H., Lurz, B., Lurz, M., & Wohlrab, J. (2023). Grundlagen der Informatik (4. Aufl.). Pearson Studium. ▪ C. Vogt (2004). Informatik – Eine Einführung in Theorie und Praxis, Spektrum. ▪ Mössenböck, H. (2014). Sprechen Sie Java (5. Aufl.). dpunkt.verlag. ▪ Ratz, D., Schulmeister-Zimolong, D., Seese, D., & Wiesenberger, J. (2022). Grundkurs Programmieren in Java (8. Aufl.). Hanser. ▪ Lo Iacono, L., Wiefeling, S., & Schneider, M. (2020). Programmieren trainieren (2. Aufl.). Hanser. ▪ Schiedermeier, R., & Köhler, K. (2012). Das Java-Praktikum (2. Aufl.). dpunkt.verlag. ▪ Ullenboom, C. (2023). Java ist auch eine Insel (17. Aufl.). Rheinwerk Verlag. Openbook.
Verwendung des Moduls in weiteren Studiengängen	P11 in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.43 PI2 - Praktische Informatik 2

Modulkürzel	PI2_BaTIN2020
Modulbezeichnung	Praktische Informatik 2
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	PI2 - Praktische Informatik 2
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	2
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Chunrong Yuan/Professor Fakultät IME
Dozierende*r	Prof. Dr. Chunrong Yuan/Professor Fakultät IME
Learning Outcome(s)	
Was: Das Modul vermittelt grundlegende Kompetenzen zur Nutzung von Programmiersprachen und entsprechender abstrakterer Darstellungsformen bei der algorithmischen und objektorientierten Lösung von Anwendungsproblemen. Die Studierenden sollen in die Lage versetzt werden, einschlägige Begrifflichkeiten und Techniken im praktischen Umfeld sicher anzuwenden: Aufbauend auf den in der Vorlesung vermittelten Kenntnissen (K3) analysieren die Studierenden Problemstellungen (K2), entwerfen dazu Lösungswege (K5), implementieren sie mit Hilfe von Standardwerkzeugen (K6, K9) und prüfen sie (K7). Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. In den Übungen und insbesondere im Praktikum bearbeiten die Studierenden in Kleingruppen Programmieraufgaben und verteidigen ihre Lösungen (K8, K13, K16) Wozu: Kompetenzen in der Anwendung von Programmiersprachen sind essentiell für Informatikerinnen und Informatiker, insbesondere im Hinblick auf die Realisierung informationstechnischer Systeme (HF1). Durch ihre praktische Programmierarbeit erwerben die Studierenden zudem Erfahrungen, die wichtig sind für die Erfassung von Anforderungen, die Entwicklung von Konzepten zur technischen Lösung und zu ihrer Bewertung (HF2). Die Durchführung im Team mit dem Dozenten als „Auftraggeber“ stärkt die Interaktionsfähigkeit der Studierenden (HF 4).	
Modulinhalte	
Vorlesung / Übungen	
Objektorientierte Programmierung: Klassenstrukturen Objektorientierte Programmierung: Generische Klassen Ausnahmeereignisse Ein-/Ausgabe: Ströme und Dateien Ein-/Ausgabe: Graphische Benutzeroberflächen (GUIs) Dynamische Datenstrukturen: Einfache Strukturen Dynamische Datenstrukturen: Graphen Formale Spezifikation syntaktischer Strukturen Objektorientierte Programmierung	
Praktikum	
Objektorientierte Implementierung von dynamischen Strukturen Objektorientierte Implementierung von GUI-Komponenten	
Lehr- und Lernmethoden	■ Vorlesung / Übungen ■ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung

Workload	150 Stunden
Präsenzzeit	45 Stunden $\triangleq$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul PI1: Grundkenntnisse in PI1 Programmiererfahrung mit Entwicklungsumgebungen wie Eclipse ▪ Grundkenntnisse in PI1 Programmiererfahrung mit Entwicklungsumgebungen wie Eclipse
Zwingende Voraussetzungen	
Empfohlene Literatur	▪ Vogt: Informatik – Eine Einführung in Theorie und Praxis, Spektrum, 2004
Verwendung des Moduls in weiteren Studiengängen	PI2 in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.44 PP - Programmierpraktikum

Modulkürzel	PP_BaTIN2020
Modulbezeichnung	Programmierpraktikum
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	PP - Programmierpraktikum
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	1
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Chunrong Yuan/Professor Fakultät IME
Dozierende*r	Prof. Dr. Chunrong Yuan/Professor Fakultät IME

Learning Outcome(s)

Was: Die Studierenden erwerben und vertiefen ihre Kompetenzen in der Analyse typischer Aufgabenstellungen (K.4) dem Entwurf (K.5), der Erstellung (K.6) und dem Test bzw. Prüfen (K.7, K.9) von Softwaresystemen und nebenbei in der Abstraktion und Formalisierung (K.2) fachlicher Probleme.

Womit: Die Studierenden erhalten ausgewählte Programmieraufgaben steigender Komplexität, die sie zunächst analysieren, dann ein passendes Programm mit vorgegebenen Methoden und modernen Entwicklungsumgebungen entwerfen und programmieren, es mit vorgegebenen und selbst festzulegenden Testfällen prüfen.

Wozu: Damit wird eine Basis gelegt, auf der sie dann später, in den Veranstaltungen höherer Semester, aber auch im Berufsleben, eigenständig IT-Aufgaben analysieren, passende Systeme entwerfen, implementieren und prüfen können.

Modulinhalte**Vorlesung**

Vorstellung ausgewählter Standardmethoden zum Entwurf und Implementation von Programmen:

- * Algorithmmierung mit Struktogramm/Programmablaufplan
- * Automaten
- * Strukturierte Analyse mit Datenflussdiagrammen und Datenverzeichnis

Benutzung einer Programmentwicklungsumgebung für Programmierung und Fehlerbeseitigung

Rekursion als Mittel zur Umsetzung von Folgen (aus der Mathematik)

Praktikum

Algorithmen

Beschreibungsformen

natürliche Sprache

grafische Darstellungsformen (Struktogramme und/oder Programmablaufpläne)

Datenflussdiagramm und Datenverzeichnis

Algorithmen zur Lösung von Standardproblemen

Iteration und Repetition

Rekursivität

Regulärer Automat

Implementierung von Algorithmen durch Programme mit Kontrollstrukturen (in Java und C) unter Benutzung von Bedingten Anweisungen (Einfach- und Mehrfach-Fallunterscheidung)

Schleifen (Iteration, Repetition)

Entwurf und Einsatz von Unterprogrammen (in Java)

insbesondere auch: Implementierung vorgegebener Schnittstellen

Umgang mit strukturierten Datentypen

Arrays

Datenverbund (in Java: Public Classes ohne Methoden)

Umgang mit einer Programmentwicklungsumgebung

Erstellung von Projekten

Debugging

Testen

Entwurf von Algorithmen zur Lösung vorgegebener Probleme

Umsetzung von Algorithmenbeschreibungen in Programme

Umgang mit einer Programmentwicklungsumgebung

Anwendung der unter "Fertigkeiten" genannten Aspekte auf praxisbezogene Szenarien durch selbstständige Arbeit in kleinem Team.

Lehr- und Lernmethoden	▪ Vorlesung
	▪ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	34 Stunden $\pm$ 3 SWS
--------------------	------------------------

Selbststudium	116 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	gleichzeitige (oder in Ausnahmefällen vorherige) Teilnahme an "Praktische Informatik 1"
-----------------------------------	---

Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: Praktikumstermine
----------------------------------	--

- | | |
|-----------------------------|--|
| Empfohlene Literatur | <ul style="list-style-type: none">▪ Online-Dokumentation der Java-Pakete (java.sun.com)▪ Online-Dokumentation der verwendeten Softwareentwicklungsumgebung (Eclipse)▪ Mössenböck, Sprechen Sie Java?, dpunkt 2011▪ Schiedermeier/Köhler, Das Java-Praktikum, dpunkt 2011▪ Vogt, Informatik, Spektrum Verlag 2004 |
|-----------------------------|--|

Verwendung des Moduls in weiteren Studiengängen	PP in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise	
------------------------------------	--

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.45 PPRA - Parallelprogrammierung und Rechnerarchitekturen

Modulkürzel	PPRA_BaTIN2020
Modulbezeichnung	Parallelprogrammierung und Rechnerarchitekturen
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	PPRA - Parallelprogrammierung und Rechnerarchitekturen
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Markus Cremer/Professor Fakultät IME
Dozierende*r	Prof. Dr. Georg Hartung/Professor Fakultät IME im Ruhestand

Learning Outcome(s)

Was: Das Modul vermittelt die Kompetenzen zur Verwendung von lose und eng gekoppelten Parallelrechnersystemen für die Erarbeitung von Problemlösungen z.B. aus dem Bereich der Simulation oder der Künstlichen Intelligenz. Die Problemlösungen werden unter Verwendung von state-of-the-art Entwicklungsumgebungen (z.B. MPI und CUDA) auf Basis gängiger Programmierparadigmen und Design-Pattern erstellt. Die Studierenden werden in die Lage versetzt, fachspezifische Begriffe, Tools und Techniken im praktischen Umfeld sicher anzuwenden. Aufbauend auf den in der Vorlesung vermittelten Kenntnissen werden komplexere Problemstellungen analysiert, auf Teilsysteme heruntergebrochen und modelliert. Darauf aufbauend wird die Problemlösung mittels Entwurfswerkzeugen implementiert und am Zielsystem in Betrieb genommen.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. Im Praktikum erarbeiten die Studierenden in Kleingruppen Problemlösungen und verteidigen diese.

Wozu: Kompetenzen in der Verwendung von parallelen Rechnersystemen sind essentiell für technische Informatiker, die im HF 1 arbeiten wollen. Durch die Entwicklung von Problemlösungen erwerben die Studierenden zudem Erfahrungen, die essentiell für das HF 2 sind. Eine projektorientierte Durchführung der Praktika in kleinen Teams mit dem Dozenten als "Auftraggeber" initiiert die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte**Vorlesung / Übungen**

- Überblick über Parallele Computerarchitekturen und Klassifikation
- Designverfahren für parallele Anwendungen (z.B. PCAM), Dekompositionsmuster (Funktionen, Datenbereiche)
- Muster zur Parallelprogrammierung, u.a. Master Worker, Single Program Multiple Data, Multiple Program Multiple Data, und Zuordnung zu Dekompositionsmustern
- Leistungsbestimmung paralleler Programme (Performance Evaluation), u.a. Amdahl'sches Gesetz
- Modellierung paralleler Programme mit Coloured Petri Nets
- Programmierung verteilter Rechnersysteme mit MPI
- Programmierung von Multicore/Parallelrechnern mit OpenMP und Rust
- Programmierung von GPU (Graphic Processing Units) mit CUDA
- Design Bereichsspezifischer Architekturen (Domain specific architectures)
- Digitale Hardware-Beschleuniger (Signalverarbeitung)
- Tensor Processing Unit (Künstliche neuronale Netze)

Projekt

- Nutzung der Parallelisierungsdesignmethoden für eine selbstgewählte Anwendung
- Modellierung einer geplanten Parallelisierung für die Anwendung
- Implementierung eines parallelen Programms für die Anwendung
- Ermittlung des Leistungsgewinns durch Parallelisierung bei der Anwendung

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Projekt
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	grundlegende prozedurale Programmierkenntnisse Grundkenntnisse in Multitasking-Programmierung Grundlegende Funktionsweise eines Von-Neumann-Rechners Grundlagen der Digitaltechnik (Automaten, Hardware-Beschreibungssprache)
-----------------------------------	--

Zwingende Voraussetzungen	Projekt erfordert Anwesenheit im Umfang von: 2 Labortermine + 1 Präsentationstermin
----------------------------------	---

Empfohlene Literatur	▪ Barlas, Gerassimos: Multicore and GPU Programming: An Integrated Approach ▪ Tanenbaum, Goodman: Computerarchitektur, Pearson Studium (Prentice Hall) ▪ Pacheco: Parallel Programming with MPI ▪ Eijkhout: Introduction to High Performance Computing
-----------------------------	---

Verwendung des Moduls in weiteren Studiengängen	PPRA in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.46 PRA - Praxisphase

Modulkürzel	PRA_BaTIN2020
Modulbezeichnung	Praxisphase
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	PRA - Praxisphase
ECTS credits	23
Sprache	deutsch und englisch
Dauer des Moduls	2 Semester
Empfohlenes Studiensemester	6,7
Häufigkeit des Angebots	jedes Semester
Modul-Verantwortliche*r	Studiengangsleiter(in) Bachelor Technische Informatik / Informatik und Systems-Engineering
Dozierende*r	verschiedene Dozenten*innen / diverse lecturers
Learning Outcome(s)	
Was: Die Studierenden sich in eine unbekannte (außeruniversitäre) Organisation einfügen und dort adäquat agieren. Womit: Die Studierenden solle ein eigenes Projekt in Abstimmung mit Kollegen und Vorgesetzten in der Firma organisieren und durchführen. Dazu müssen sie selbständig wissenschaftliche Methoden und moderne Technologien in der Praxis anwenden und sich selbständig in ein neues Aufgabengebiet einarbeiten. Typischerweise wird ein System oder eine Systemkomponente selbständig und im Team konzeptionieren und entwickeln. Wozu: Die Studierenden solle einen ersten Einblick in ihr zukünftiges Berufsumfeld und die damit verbundenen fachlichen aber auch sozialen Gegenbenheiten bekommen.	
Modulinhalte	
externes Praktikum	
markdown - Einblick in die Arbeitsabläufe und Organisationsform von Firmen gewinnen. - Aufgaben und Tätigkeiten der späteren Berufswelt nach dem Studium einschätzen. markdown - Eigenes Projekt in Abstimmung mit Kollegen und Vorgesetzten in der Firma organisieren und durchführen. - Selbständig wissenschaftliche Methoden und moderne Technologien in der Praxis anwenden und einsetzen. - Selbständige Einarbeitung in ein neues Aufgabengebiet. - Sich in eine unbekannte (außeruniversitäre) Organisation einfügen und dort adäquat agieren. - Eine System oder eine Systemkomponente selbständig und im Team konzeptionieren und/oder entwickeln und/oder testen. - Arbeitsergebnisse schriftlich und mündlich präsentieren.	
Lehr- und Lernmethoden	externes Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	690 Stunden
Präsenzzeit	0 Stunden $\triangleq$ 0 SWS
Selbststudium	690 Stunden
Empfohlene Voraussetzungen	
Zwingende Voraussetzungen	■ externes Praktikum erfordert Anwesenheit im Umfang von: Mit Praktikumsgeber zu vereinbaren ■ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an externes Praktikum

Empfohlene Literatur

Verwendung des Moduls in weiteren Studiengängen	PRA in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise	Das Spektrum der konkreten Aufgabenstellung umfasst die gesamte Thematik der Informatik und angrenzender Ingenieurwissenschaften. Von daher sind alle im vorangestellten Studienverlaufs vermittelten Kompetenzen potentiell notwendig und lassen sich allein durch die konkrete Aufgabenstellung, die während der Praxisphase bearbeitet wird, individuell begrenzen.
------------------------------------	--

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.47 PUK - Präsentation und Kommunikation

Modulkürzel	PUK_BaTIN2020
Modulbezeichnung	Präsentation und Kommunikation
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	PUK - Präsentation und Kommunikation
ECTS credits	3
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	5
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Studiengangsleiter(in) Bachelor Technische Informatik / Informatik und Systems-Engineering
Dozierende*r	NN/Lehrbeauftragter
Learning Outcome(s)	
Die Studierenden können moderne Präsentations- und Argumentationstechniken anwenden, indem Sie - Typen von Kommunikationsprozessen bestimmen, - Konzepte zur Vermittlung von Kommunikationsinhalten einordnen, - Grundzüge der Rhetorik sowie - moderne Präsentations- und Moderationstechniken anwenden, um in begleitenden und späteren Modulen, sowie im späteren Berufsleben technische und wissenschaftliche Themen zielgruppengerecht präsentieren, technische-wissenschaftliche Texte verfassen und sachgerecht argumentieren können.	
Modulinhalte	
Seminar	
- Typen von Kommunikationsprozessen - Konzepte zur Vermittlung von Kommunikationsinhalten - Grundzüge der Rhetorik - moderne Präsentationstechniken - Moderation von Gruppen	
Lehr- und Lernmethoden	Seminar
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	90 Stunden
Präsenzzeit	12 Stunden $\pm$ 1 SWS
Selbststudium	78 Stunden
Empfohlene Voraussetzungen	■ Modul SYP: SYP ist keine direkte Voraussetzung, aber die in PUK vermittelten Kompetenzen werden im Rahmen von SYP bei Projektpräsentation geprüft. ■ keine
Zwingende Voraussetzungen	Seminar erfordert Anwesenheit im Umfang von: 3 Termine
Empfohlene Literatur	

Verwendung des Moduls in weiteren Studiengängen	PUK in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise	
------------------------------------	--

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.48 QKC - Quellen- und Kanalcodierung

Modulkürzel	QKC_BaTIN2020
Modulbezeichnung	Quellen- und Kanalcodierung
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	QKC - Quellen- und Kanalcodierung
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Uwe Dettmar/Professor Fakultät IME
Dozierende*r	Prof. Dr. Uwe Dettmar/Professor Fakultät IME

Learning Outcome(s)

Was? Die in gespeicherten oder aktuell entstehenden Daten befindliche Information extrahieren und gegen Fehler bei Übertragung über einen gestörten Kanal und Abhören durch Dritte schützen und zugehörige Verfahren analysieren und bewerten.
Womit? Durch Anwendung von Verfahren und Algorithmen der Quellen- und Kanalcodierung und der Kryptographie.
Wozu? Zur Gewährleistung einer vertraulichen, effizienten und sicheren Speicherung und Übertragung von Daten mit Hilfe von Nachrichtentechnischen Systemen.

Modulinhalte**Vorlesung / Übungen**

Vorlesung und Übungen werden in einer Lehrveranstaltung kombiniert. Nach der Vorstellung von neuem Lernstoff durch den Dozenten in Form von kurzen Blöcken wird dieser direkt von den Studierenden durch kurze Matlab- und Python-Übungen angewendet und vertieft. Längere Übungsaufgaben werden bereits zu Hause vorbereitet und die verschiedenen Lösungsvorschläge in der Präsenzveranstaltung besprochen.

Über ein Lernportal werden elektronische Minitests zum aktuell behandelten Stoff als weitere Lernressource angeboten.

Inhalt:

- Vermittlung von Grundprinzipien und -begriffen
- Systemtheoretische Beschreibung eines kommunikationstechnischen Übertragungssystems
- mathematische Grundlagen zur Quellen- und Kanalcodierung und der Kryptographie
- Informationstheoretische Aspekte der Quellen- und Kanalcodierung
- Praktische Codes zur Quellen- und Kanalcodierung
- Aspekte der Informationssicherheit
- public und private key Kryptographie und praktische Anwendung
- kryptologische Protokolle

Die Studierenden lernen die o.g. Themen in der Vorlesung kennen, erwerben Grundwissen und vertiefen dieses durch Selbststudium mit Hilfe von Literatur, YouTube Videos und anderen Netzressourcen (selbstständige Informationsbeschaffung), sowie in Lerngruppen (Teamwork).

Durch kleine Übungsaufgaben und Programme wird in der Präsenzveranstaltung bereits ein aktiver Umgang mit den vorgestellten Verfahren ermöglicht. Umfangreichere Rechenaufgaben werden am Ende der Veranstaltung behandelt und die Lösungswege diskutiert, um dadurch den Studierenden relevante Problemstellungen vorzustellen und ihre Fähigkeit zur Lösungsfindung zu entwickeln.

Die Studierenden lernen darüber hinaus:

- nachrichtentechnische Systeme zu analysieren und deren Performanz zu ermitteln bzw. abzuschätzen.
- Verfahren der Quellen- und Kanalcodierung und Kryptologie zu vergleichen und zu bewerten
- Kenntnisse auf technische Problemstellungen anzuwenden

Praktikum

Bearbeitung von geeigneten Praktikumsaufgaben aus dem Bereich der Quellen- und Kanalcodierung in Form von Jupyter Notebooks. Die Studierenden verwenden dabei teiltfertige oder vorhandene Programme für Simulationen. Sie notieren die Ergebnisse, erzeugen graphische Darstellungen und diskutieren die Ergebnisse.

Matlab mit der Communications Toolbox wird für Simulationsaufgaben verwendet, deren zeitlicher Aufwand für eine Eigenentwicklung zu groß ist.

- Die Studierenden schulen ihre Fähigkeiten zur Lösung technischer Probleme mit Hilfe von Computerprogrammen.
- Sie analysieren und simulieren nachrichtentechnische Systeme und bewerten deren Eigenschaften.
- Sie schulen ihre Selbstorganisation und ihr problemorientiertes Denken und Handeln.
- Sie trainieren das Lösen von Aufgaben im Team und ihre kommunikativen Fähigkeiten.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	■ Modul MA1: Kenntnisse in Lineare Algebra, Funktionentheorie, Algebra Fähigkeit, diese Kenntnisse in praktischen Problemen anzuwenden. ■ Modul MA2: Kenntnisse: Reihen und Folgen, Fehlerrechnung Fähigkeit, diese Kenntnisse in praktischen Problemen anzuwenden. ■ Modul PI1: Algorithmen zur Lösung vorgegebener Probleme formulieren Beherrschung grundlegender Programmierfähigkeiten ■ Die Studierenden sollten Grundkenntnisse in den Gebieten Lineare Algebra, Stochastik und Algebra und zusätzlich Programmierkenntnisse mitbringen, die es Ihnen ermöglichen, einfache Programme in einer höheren Programmiersprache zu schreiben. In der Vorlesung werden Matlab/Octave und Python verwendet.
Zwingende Voraussetzungen	■ Praktikum erfordert Anwesenheit im Umfang von: 4 Praktikumstermine ■ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	■ BOSSERT, M. : Einführung in die Nachrichtentechnik. Oldenbourg Verlag, 2012. ■ BOSSERT, M. : Kanalcodierung. Oldenbourg Wissenschaftsverlag GmbH, 2013. ■ NEUBAUER, A. : Informationstheorie und Quellencodierung. Wilburgstetten : Schönbach, 2006. ■ PROAKIS, J. G. ; SALEHI, M. : Digital Communications. 5. McGraw-Hill, 2008. ■ SAYOOD, K. : Introduction to data compression. third. Elsevier Morgan Kaufmann, 2000. ■ MEYER, M. : Kommunikationstechnik. 4. Vieweg und Teubner, 2019. ■ SKLAR, B. : Digital Communications. Prentice Hall PTR, 2001
Verwendung des Moduls in weiteren Studiengängen	■ QKC in Bachelor Elektrotechnik 2020 ■ QKC in Bachelor Elektrotechnik und Informationstechnik 2024 ■ QKC in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.49 RT - Regelungstechnik

Modulkürzel	RT_BaTIN2020
Modulbezeichnung	Regelungstechnik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	RT - Regelungstechnik
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Jens Onno Krah/Professor Fakultät IME
Dozierende*r	Prof. Dr. Jens Onno Krah/Professor Fakultät IME

Learning Outcome(s)

Statisches Verhalten von Regelstrecken und Regelkreisen analysieren.
 Üben anhand von Kennlinienfeldern und Linearisierungen.
 Dynamisches Verhalten von Regelstrecken kennenlernen
 Empirische Betrachtungen durchführen, Differentialgleichungen aufstellen, Laplace-Transformation verwenden, Übertragungsfunktionen berechnen, Frequenzgang und Bode-Diagramm erstellen.
 Stabilität von Regelkreisen
 Algebraische Stabilitätskriterien anwenden, Nyquist-Kriterium verwenden.
 Parametrierung von Reglern
 Anwenden von Entwurfsverfahren, Entwerfen mit Frequenzkennlinien / Bode-Diagramm, Parametrieren durch Polvorgabe
 Gerätetechnik, zeitdiskreter Regelkreis
 Kennlernen von dedizierten Reglern und Differenzengleichungen
 Algorithmische Abtastregelungen parametrieren.
 Vermaschte Regelkreise
 Kennlernen von Kaskadenregelung, optional mit Vorsteuerung bzw. Störgrößenaufschaltung.
 Split-Range-Regelungen anwenden.
 Technisches Englisch
 Beispielsweise Fachgespräche sollen auf Englisch geführt werden können.

Modulinhalte

Vorlesung / Übungen

Übertragungsfunktion des geschlossenen Regelkreises
 Wahl eines geeigneten Reglers bei gegebener Strecke
 Berechnung der Stabilität von Regelkreisen

Praktikum

Handhabung und korrekte Anwendung von Simulationswerkzeugen
 Einsatz und Beurteilung der Funktion von Reglern
 Aufbau von Regeleinrichtungen
 Lösung von Regelaufgaben

Lehr- und Lernmethoden

- Vorlesung / Übungen
- Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\triangleq$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	MA1, MA2, GE1, GE2, ASS, MT
Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: 3 Praktikumstermine
Empfohlene Literatur	▪ Skript ▪ Lutz, Wendt: Taschenbuch der Regelungstechnik, Verlag Harri Deutsch.
Verwendung des Moduls in weiteren Studiengängen	▪ RT in Bachelor Elektrotechnik 2020 ▪ RT in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ RT in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.50 SE - Software Engineering

Modulkürzel	SE_BaTIN2020
Modulbezeichnung	Software Engineering
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	SE - Software Engineering
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	3
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Hans Nissen/Professor Fakultät IME
Dozierende*r	Prof. Dr. Hans Nissen/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt Kompetenzen zur Erstellung von Software-Systemen in allen Projektphasen von der Dokumentation von Systemanforderungen (K.1, K.4), der Spezifikation und Modellierung von Systemeigenschaften (K.1, K.2, K.3, K.5, K.9), der Erstellung und Bewertung eines Systementwurfs für vorgegebene Qualitätsziele (K.1, K.2, K.5, K.9, K.10), der Prüfung von Systemeigenschaften (K.7, K.9, K.10) und der Erstellung leserlichen Programmcode (K.6). Im parallel laufenden Praktikum werden die Kompetenzen zur Erstellung umfangreicher technischer Texte (K.2, K.4), zum Entwurf eines Software-Systems unter Berücksichtigung von Qualitätszielen (K.1, K.3, K.5) zur Realisierung von Systemmodellen in Programmcode (K.6), zur Prüfung von Programmcode (K.6, K.7), zur Prüfung eines erstellten Software-Systems (K.7, K.9, K.10) und die Fähigkeit, komplexe fachbezogene Probleme zu sehen und Lösungen gegenüber Fachleuten mündlich argumentativ zu vertreten und mit anderen Studierenden weiterzuentwickeln (K.3, K.5, K.7, K.9, K.11, K.16) verstärkt.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs/Übungsteil unter Einbeziehung einer umfangreichen Fallstudie und betreut parallel dazu ein Praktikum, in dem die Studierenden ein kleines Software-System planen, entwerfen, realisieren und analysieren.

Wozu: Kompetenzen in der Entwicklung eines Software-Systems sind essentiell für technische Informatiker, die im HF 1 arbeiten wollen. Durch die Arbeit an einem komplexeren Beispielsystem erwerben die Studierenden zudem Erfahrungen, die essentiell für das HF 2 sind, u.a. Anforderungen spezifizieren, Systeme und Software-Architekturen entwerfen, realisieren und bewerten. Die Entwicklung eines Software-Systems über mehrere Praktikumstermine hinweg, vermittelt den Studierenden erste Erfahrungen in der Organisation umfangreicherer Systeme unter Berücksichtigung vorgegebener Qualitätsziele (HF 3).

Modulinhalte***Vorlesung / Übungen***

Aufgaben und Disziplinen des Software Engineering
 Vorgehensmodelle
 Aufgaben, Methoden und Techniken des Anforderungsmanagement
 unterschiedliche Techniken zur System- und Software-Spezifikation
 grundlegende Modellierung in UML
 moderne Architekturstile kennen und bewerten können
 Methoden der Qualitätssicherung
 Aufgaben, Methoden und Techniken des Konfigurationsmanagement
 Dokumentation von Anforderungen
 Bewertung von Vorgehensmodellen
 Erstellung von Systemmodellen
 Erstellung und Bewertung alternativer System-Architekturen
 Erstellung und Bewertung alternativer Software-Architekturen
 Ableitung geeigneter logischer und konkreter Testfälle
 Erstellung eines lesbaren Programmcodes

Praktikum

umfangreichen Text verstehen
 Verwendung von Modellierungswerkzeugen
 Erstellung korrekter Modelle
 Programme in objektorientierter Sprache (Java) erstellen
 Prüfung von Programmen
 gegebenes Modell in Programmcode übersetzen
 Systemmodelle aus gegebenem Lastenheft ableiten
 Systementwurf zu Systemmodellen erstellen
 Implementierung der Systemmodelle
 gegebenes Programm prüfen

Lehr- und Lernmethoden	■ Vorlesung / Übungen ■ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\triangleq$ 4 SWS
--------------------	-------------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	■ Modul PI2: Fundierte Kenntnisse in der Programmierung. ■ Modul PP: Praktische Erfahrungen in der Programmierung. ■ Programmierkenntnisse in Java
-----------------------------------	--

Zwingende Voraussetzungen	■ Praktikum erfordert Anwesenheit im Umfang von: 3 Testattermine ■ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
----------------------------------	---

Empfohlene Literatur	■ I. Sommerville: Software Engineering, Addison-Wesley, 2018. ■ H. Balzert: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, Spektrum Akademischer Verlag, 3. Auflage, 2009. ■ B. Oestereich: Analyse und Design mit der UML 2.5: Objektorientierte Softwareentwicklung, Oldenbourg Verlag, 11. Auflage, 2013. ■ B. Brügge, A.H. Dutoit: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java, Pearson Studium, 2006. ■ H. Balzert: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, Spektrum Akademischer Verlag, 3. Auflage, 2012. ■ T. Posch, K. Birken, M. Gerdorf: Basiswissen Softwarearchitektur, 3. Auflage, dpunkt Verlag, 2011. ■ A. Spillner, T. Linz: Basiswissen Softwaretest, 4. Auflage, dpunkt Verlag, 2010. ■ B. Hindel, K. Hörmann, M. Müller, J. Schmied: Basiswissen Software-Projektmanagement, 3. Auflage, dpunkt Verlag, 2009. ■ G. Popp: Konfigurationsmanagement, 3. Auflage, dpunkt Verlag, 2009. ■ R. Oechsle: Java-Komponenten Grundlagen, prototypische Realisierung und Beispiele für Komponentensysteme, Hanser Verlag, 2013.
-----------------------------	---

Verwendung des Moduls in weiteren Studiengängen	SE in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.51 SIG - Signalverarbeitung

Modulkürzel	SIG_BaTIN2020
Modulbezeichnung	Signalverarbeitung
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	SIG - Signalverarbeitung
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	3
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Harald Elders-Boll/Professor Fakultät IME
Dozierende*r	Prof. Dr. Harald Elders-Boll/Professor Fakultät IME

Learning Outcome(s)

Die Studierenden lernen die grundlegenden Verfahren und Algorithmen zur Analyse und Verarbeitung von diskreten Signalen und Systemen im Zeit- und Frequenzbereich anzuwenden, wie die diskrete Faltung, die DTFT, die z-Transformation und die DFT/FFT, die Eigenschaften zeitdiskreter Signale und Systeme im Zeit- und Frequenzbereich zu ermitteln, darzustellen und zu interpretieren, um analoge Signale digitalisieren, analysieren und mit Hilfe von zeitdiskreten Systemen verarbeiten zu können.

Modulinhalte**Vorlesung / Übungen**

Grundbegriffe: Klassifikation von Signalen und Systemen, Stabilität, Kausalität,
 LTI- und LSI-Systeme: Impulsantwort, Faltung, Stabilität, Kausalität, Korrelationsfunktionen
 Fourier-Reihe: reelle und komplexe Koeffizienten, Gibbs'sches Phänomen
 Fourier-Transformation: Eigenschaften und Theoreme, Berechnung, Übertragungsfunktion, Energiedichtespektrum
 Abtastung: abgetastete und zeitdiskrete Signale, Abtasttheorem, Aliasing
 DTFT: Herleitung, Korrespondenzen und Theoreme, Berechnung, Frequenzgang
 DFT: Herleitung, Korrespondenzen und Theoreme, Leakage-Effekt

Beurteilung der Stabilität von LTI- und LSI-Systemen im Zeitbereich
 Berechnung des Ausgangssignals von LSI-Systemen mit Hilfe der Faltung
 Berechnung der Fourier-Transformation und der DTFT
 Implementierung von FIR Systemen durch Programmierung der diskreten Faltung
 Implementierung von einfachen IIR Systemen
 Beurteilung der Filtercharakteristik anhand des Frequenzgangs und des Höreindrucks

Praktikum

Zwei Laborversuche zur digitalen Signalverarbeitung akustischer Signale am Rechner mit iPython Notebooks, um die in der Vorlesung/Übung erworbenen Kenntnisse und Fertigkeiten praktisch anzuwenden:

1. Zeitdiskrete Signale und Systeme im Zeitbereich:
 Programmierung der zeitdiskreten Faltung und Implementierung von einfachen FIR Filtern
 Programmierung eines einfachen rekursiven (IIR) Systems
 Beurteilung der Wirkung der Filter anhand von akustischen Signalbeispielen
2. Die Fourier-Reihe
 Animation der Signalsynthese periodischer Signale als Summe von Sinus- und Cosinus-Signalen für folgende Beispiele:
 Rechteck-Schwinung, Dreieck-Schwinung, Sägezahn,
 Logarithmische Darstellung des Spektrums
 Bestimmung des Klirrfaktors
 Relation zwischen Klang und Klirrfaktor

Einfache Algorithmen der Signalverarbeitung

Lehr- und Lernmethoden	■ Vorlesung / Übungen ■ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	■ Modul MA1: trigonometrische, exp., log-Funktionen; Grenzwerte; komplexe Rechnung ■ Modul MA2: Integral- und Differentialrechnung; unendliche Reihen; Reihenentwicklung ■ Modul PI1: Prozedurale Programmierung ■ Kenntnisse der folgenden mathematischen Grundlagen: trigonometrische, exp., log-Funktionen; Grenzwerte; komplexe Rechnung, Integral- und Differentialrechnung; unendliche Reihen; Partialbruchzerlegung; Reihenentwicklung
-----------------------------------	---

Zwingende Voraussetzungen	■ Praktikum erfordert Anwesenheit im Umfang von: 2 Praktikumstermine ■ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
----------------------------------	---

Empfohlene Literatur	■ Jens Rainer Ohm und Hans Dieter Lüke, Signalübertragung, Springer, 2014 ■ Martin Meyer, Signalverarbeitung, Springer Vieweg, 2014 ■ Martin Werner, Signale und Systeme, Springer Vieweg, 2008 ■ Bernd Girod u.a., Einführung in die Systemtheorie, Springer Vieweg, 2007
-----------------------------	---

Verwendung des Moduls in weiteren Studiengängen	SIG in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise	
------------------------------------	--

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.52 SM - Software Management

Modulkürzel	SM_BaTIN2020
Modulbezeichnung	Software Management
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	DOPE - DevOps und Platform-Engineering
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. René Wörzberger/Professor Fakultät IME
Dozierende*r	Prof. Dr. René Wörzberger/Professor Fakultät IME

Learning Outcome(s)

(WAS) Die Studierenden erlernen und wenden Werkzeuge und Methoden an zur (a) professionellen Entwicklung von Systemen im Team, (b) zur Qualitätssicherung von Systemen, (c) zur Automatisierung von Entwicklungsprozessen und (d) zum Betrieb von Systemen in Cloud-Infrastrukturen ,
(WOMIT) indem ihnen besagte Inhalte und Fertigkeiten in Vorlesungen/Übungen vermittelt werden und indem sie sie in einer Reihe von vorzubereitenden, aufeinander aufbauenden Praktikumsaufgaben praktisch anwenden,
(WOZU) um später in hochdynamischen Entwicklungsumfeldern qualitativ hochwertige Software-Systeme über den gesamten Lebenszyklus verantworten zu können.

Modulinhalte**Vorlesung / Übungen**

Interne Funktionsweise des Versionsverwaltungssystems Git
 Team-Organisation mit Funktionalitäten von GitLab
 Build-Automatisierung mit Apache Maven
 Continuous-Integration and -Delivery (CI/CD) mit GitLab-Runner
 Automatisierung von Tests mit JUnit
 Erstellung von Mocks mit Mockito
 Automatisierung von WebUI-Tests mit Selenium
 Automatisierung von Lasttests mit Apache JMeter
 Vermessung von Code-Qualität mit Sonarqube
 Klassische und Cloud-Infrastrukturen
 Erstellung eines System-Clusters in der Google Cloud
 Container-Virtualisierung mit Docker
 Container-Orchestrierung mit Kubernetes

Praktikum

Entwickeln im Team mit GitLab
 Einpflegen und Weiterentwickeln der Code-Basis in/mit Git
 Erstellung von Build-Skripts mit Maven
 Implementieren von Tests mit JUnit, Mockito, Selenium und JMeter
 Containerisierung und Deployment mit Docker und Kubernetes
 Aufbau eines System-Clusters in der Google Cloud inklusive (kontinuierlichem) Deployment von Releases in diese.

Lehr- und Lernmethoden	■ Vorlesung / Übungen
	■ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	■ Modul PI1: Ein fortgeschrittenes Verständnis des "gemanagten" Subjekts im Java-Source-Code ist erforderlich.
	■ Modul PI2: Ein fortgeschrittenes Verständnis des "gemanagten" Subjekts im Java-Source-Code ist erforderlich.
	■ Modul SP: Ein fortgeschrittenes Verständnis des "gemanagten" Subjekts im Java-Source-Code ist erforderlich. Zudem sind für die Bearbeitung von Praktikumsaufgaben fortgeschrittene Kenntnisse mit der Arbeit im Team erforderlich.
	■ (1) fortschrittene Kenntnisse in der Programmierung in Java (2) Erfahrungen mit der Entwicklung im Team (3) Kenntnisse in Software-Engineering

Zwingende Voraussetzungen

Empfohlene Literatur	■ H. Atkins et al.: Building Secure and Reliable Systems, O'Reilly, 2020 ■ B. Betsy et al.: Site Reliability Engineering, O'Reilly, 2020 ■ J. Humble et al.: Accelerate - The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations, IT Revolution, 2018
-----------------------------	---

Verwendung des Moduls in weiteren Studiengängen	■ SM in Bachelor Elektrotechnik 2020 ■ DOPE in Bachelor Informatik und Systems-Engineering 2024
--	---

Perma-Links zur Organisation	<u>Ilu-Kurs</u>
-------------------------------------	---------------------------------

Besonderheiten und Hinweise	
------------------------------------	--

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.53 SMO - Smart Mobility Components

Modulkürzel	SMO_BaTIN2020
Modulbezeichnung	Smart Mobility Components
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	SMO - Smart Mobility Components
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Kai Kreisköther/Professor Fakultät IME
Dozierende*r	Prof. Dr. Kai Kreisköther/Professor Fakultät IME

Learning Outcome(s)

WAS

Die Studierenden können

- konzeptionelle und technologische Entscheidungen für Komponenten und Gesamtsystemarchitekturen von smarten Mobilitäts- und Logistiksystemen treffen,

WOMIT

indem sie

- Ausprägungen und Bausteine von smarten Mobilitäts- und Logistiksystemen kennenlernen,
- die Vielfalt der Komponenten in smarten Mobilitäts- und Logistiksystemen und deren prinzipielle technologische Umsetzung kennenlernen,
- im Rahmen einer Projektarbeit eigenständig eine technologische Komponente eines smarten Mobilitäts- oder Logistiksystems konzipieren, spezifizieren und entwickeln/testen,

WOZU

um

- im Rahmen der konzeptionellen und technologischen Gestaltung von smarten Mobilitäts- und Logistiksystemen als mündiges Projektmitglied auftreten und mitarbeiten zu können und
- technologische Komponenten (bspw. Sensoren) sowie Gesamtsystemarchitekturen von smarten Mobilitäts- und Logistiksystemen konzipieren, spezifizieren und entwickeln/testen zu können.

Modulinhalte**Vorlesung / Übungen**

- Motivation für Smart Mobility und Logistik
- Überblick über Formen für Smart Mobility und ihre (mobilen und stationären) Komponenten (bspw. Carsharing, Bikessharing, Scooter-Sharing, Telematik im ÖPNV, Smarte Mobilitäts-Infrastruktur)
- Überblick über Formen für Smart Logistik und ihre (mobilen und stationären) Komponenten (bspw. für Langstrecken-Begegnungsverkehr)
- Technologische Komponenten der Smart Mobility und Logistik (insb. Sensoren und Sensorsysteme)

Projekt

- In einer das Semester begleitenden Projektarbeit die im Rahmen der Vorlesung erlangten Kenntnisse anwenden, vertiefen und reflektieren.
- Ein Smart Mobility System (Sensor, Serverkomponente, o.ä.) auswählen und prototypisch konzipieren und implementieren (Elektronik/Hardware, Software)

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Projekt
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	Systementwurf, Elektronische Schaltungen (Microcontroller, Sensoren, ...), Programmierung (C/C++, Java, Python), Technisches Projektmanagement
Zwingende Voraussetzungen	Projekt erfordert Anwesenheit im Umfang von: 8 Review-Termine á 90 Minuten
Empfohlene Literatur	▪ Barbara Flügge: Smart Mobility, 2020 ▪ Alaa Khamis: Smart Mobility, 2021 ▪ Fateh Belaïd und Anvita Arora: Smart Cities, 2024 ▪ Michèle Finck et al: Smart Urban Mobility, 2020 ▪ Vincenzo Piuri et al: AI and IoT for Smart City Applications, 2022 ▪ Dagmar Cagáňová und Natália Horňáková: Industry 4.0 Challenges in Smart Cities, 2023
Verwendung des Moduls in weiteren Studiengängen	▪ SMO in Bachelor Elektrotechnik 2020 ▪ SMO in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ SMO in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 15:19:39

6.54 SMP - Signalverarbeitung mit Matlab/Python und μ C

Modulkürzel	SMP_BaTIN2020
Modulbezeichnung	Signalverarbeitung mit Matlab/Python und μ C
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	SMP - Signalverarbeitung mit Matlab/Python und μ C
ECTS credits	5
Sprache	deutsch und englisch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. Harald Elders-Boll/Professor Fakultät IME
Dozierende*r	▪ Prof. Dr. Harald Elders-Boll/Professor Fakultät IME ▪ Prof. Dr. Uwe Dettmar/Professor Fakultät IME ▪ Prof. Dr.-Ing. Christoph Pörschmann/Professor Fakultät IME

Learning Outcome(s)

Entwurf, Analyse und Implementierung von Systemen und Algorithmen zur Signalverarbeitung in Software und Hardware durch praktische Übungen und das selbstständige Bearbeiten von Hard- und/oder Software-Projekten, um erfolgreich neue Systeme und Anwendungen der Signalverarbeitung in unterschiedlichen Anwendungsbereichen entwickeln zu können

Modulinhalte**Vorlesung**

Prinzipien der digitalen Signalverarbeitung:
Abtastung und Rekonstruktion
Digitale Filter
DFT und FFT
Implementierung der Faltung mit Hilfe der FFT
Spektralanalyse
Signalgenerierung

Echtzeitsignalverarbeitung:
Interrupt und Polling
Blockbasierte Signalverarbeitung

Grundlagen der digitalen Signalverarbeitung anwenden:
Grundlegende Prinzipien der digitalen Signalverarbeitung verstehen und erklären können
Unterschiedliche Filter Typen und Implementierungen vergleichen und bewerten können

Implementierung und Echtzeitsignalverarbeitung:
Grundlegende Problematik der Echtzeitsignalverarbeitung darstellen können
Einflussfaktoren auf die Verarbeitungsgeschwindigkeit benennen können
Grundlegende Verfahren zur Echtzeitsignalverarbeitung verstehen und erklären können

Praktikum

Implementierung einfacher Verfahren der Signalverarbeitung in Python/Matlab und auf Mikroprozessoren.

Projekt

Implementierung in Python/Matlab:
Algorithmus in Python/Matlab programmieren, debuggen und optimieren.

Implementierung auf einem Mikroprozessor
Algorithmus in C auf Zielprozessor programmieren
Entwicklungsumgebung kennen und nutzen können
Algorithmus auf den verwendeten Hardware effizient realisieren

komplexe Aufgaben im Team bewältigen:
einfache Projekte planen und steuern
Absprachen und Termine einhalten
Reviews planen und durchführen

Verfahren der Signalverarbeitung auf Zielplattform implementieren:
Vorgegebene Verfahren der digitalen Signalverarbeitung verstehen
Notwendige Literatur beschaffen und verstehen
Mathematisch formulierte Verfahren in Programmcode umsetzen
Programm testen, prüfen und optimieren

Arbeitsergebnisse darstellen:
Präsentation der Ergebnisse der Projektarbeit (in Englisch)

Lehr- und Lernmethoden	▪ Vorlesung ▪ Praktikum ▪ Projekt
-------------------------------	---

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul SIG: Grundbegriffe von zeitdiskreten Signalen und Systemen, Stabilität, Kausalität LSI-Systeme: zeitdiskrete Faltung zeitdiskreter Signale, FIR und IIR Filter Abtastung, Abtasttheorem, Aliasing DTFT, Frequenzgang z-Transformation, Zusammenhang zwischen Frequenzgang und Übertragungsfunktion, Blockschaltbilder DFT, Leakage-Effekt ▪ grundlegende prozedurale Programmierkenntnisse Grundkenntnisse der digitalen Signalverarbeitung: Abtasttheorem, Digitale Filter, Fouriertransformation
Zwingende Voraussetzungen	▪ Projekt erfordert Anwesenheit im Umfang von: 8 Termine ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Projekt
Empfohlene Literatur	▪ Welch, Wright, Morrow: Real-Time Digital Signal Processing (CRC Press)
Verwendung des Moduls in weiteren Studiengängen	▪ SMP in Bachelor Elektrotechnik 2020 ▪ SMP in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ SMP in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.55 SOP - Systems on Programmable Chips

Modulkürzel	SOP_BaTIN2020
Modulbezeichnung	Systems on Programmable Chips
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	SOP - Systems on Programmable Chips
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Tobias Krawutschke/Professor Fakultät IME
Dozierende*r	Prof. Dr. Tobias Krawutschke/Professor Fakultät IME

Learning Outcome(s)

Die Studierenden erwerben die Kompetenz zum Entwurf, Implementierung und Test eines modernen signalverarbeitenden Systems, indem sie an einfachen Beispielen die FPGA-Technologie mittels Hardware-Beschreibungssprache benutzen lernen, dies dann auf eine komplexere Aufgabenstellung aus der Audio-Signalverarbeitung anwenden, damit sie später FPGAs als "Problemlöser" für leistungsfähige Verarbeitung von Signalen einsetzen können.

Die Studierenden erwerben die Kompetenz zum Entwurf eines Hardware-Software-Systems, indem sie auf der Basis ihrer Kenntnisse in hardwarenaher Programmierung und der Erstellung programmierter digitaler Systeme ein Beispielsystem auf einem SoPC (System on Programmable Chip) erstellen, damit sie später diese Technologie für verschiedenste Aufgaben, bei denen viele Daten in kürzester Zeit bearbeitet werden müssen, anwenden können.

Modulinhalte**Vorlesung / Übungen**

- 1) Digitaltechnische Systeme beschreiben (modellieren) mittels
 - Boole'scher Algebra
 - Schaltplan aus existierenden Bausteinen
 - Endlichen Automaten (Zustands-Übergangs-Diagramme)
 - Erweiterte Automaten und Statecharts
 - Kontrollfluss-Datenflusssysteme
 - VHDL
- 2) Digitale Technologie
 - Typische Schaltungen (CMOS) in ihrem Verhalten verstehen und beschreiben
 - Laufzeiteffekte in Schaltnetzen verstehen, beschreiben und klassifizieren
 - Aufbau und Funktionsweise programmierbarer Bausteine verstehen und beschreiben
- 3) SoC/SoPC-Systeme
 - Systemaufbau
 - IO-Zugriffe über maschinennahe Programmierung
 - Interrupts und Alarme
 - Programmierung Automatensteuerung/CFDF-System
 - Regeln für Hardware/Softwareaufteilung
 - Design der Kopplung von HW/SW-Komponenten

Praktikum

Erwerb von Kompetenzen in der Analyse, Modellierung und Umsetzung des Hardware-Teils eines audioverarbeitenden Systems:

- 1) Analyse der Schnittstelle zum vorgegebenen CoDec und Erstellung einer Kopier-Hardware zum Einlesen und Ausgeben der Samples
- 2) Aufbau eines FIR Filters für die Samples
- 3) Aufbau einer simplen Echo-Erzeugungseinheit (Arbeit im Zeitbereich)

Erwerb von Kompetenzen in der Analyse, Modellierung und Umsetzung eines audioverarbeitenden Systems in Software:

- 1) Analyse der Schnittstelle zum vorgegebenen CoDec und Erstellung einer Kopier-Software zum Einlesen und Ausgeben der Samples
- 2) Aufbau eines N-stufigen Average-Mean-Filters für die Samples
- 3) Aufbau einer simplen Echo-Erzeugungseinheit (Arbeit im Zeitbereich)
- 4) Messung, Optimierung des Systems, das an der Leistungsgrenze üblicher Mikrocontroller arbeitet

Realisation des Beispielsystems als HW/SW-System mit einstellbaren Parametern für Echo und FIR-Filter

- 1) Aufgabenaufteilung HW/SW
- 2) Festlegung des Protokolls zwischen HW- und SW-Komponenten
- 3) Realisierung des User Interface (Eingabe der Echo- und Filterparameter, Steuerung des Systems)
- 4) Realisierung der Protokoll-Komponenten
- 5) Validierung mit FPGA-Board
- 6) Vergleich der Lösungen HW / SW / SoPC in einem Bericht

Lehr- und Lernmethoden	■ Vorlesung / Übungen
	■ Praktikum

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul DR: Grundlagen Digitale Logik Grundlagen Automaten Grundlagen Mikroprozessor Grundlagen Hardwarenahe Programmierung in C ▪ Modul PP: Programmier-Kompetenzen Kompetenz zur Textanalyse und Extraktion der Informationen für einen Programmwurf Strukturierte Analyse ▪ Modul BVS1: Konzepte des Multitasking ▪ Grundwissen Digitalrechner * Beschreibungsformen Digitaltechnik (Boole'sche Algebra, Automaten) * Grundkenntnisse digitale Technologie inkl. HDL (Hardware description language) ▪ Grundwissen Programmierung * Hardwarenahe Programmiersprache C * Programmiererfahrung * Kenntnisse und Anwendungserfahrung von Konzepten für reaktiver Programmierung, insb. Interrupts ▪ Grundwissen Signalverarbeitung, insb. Diskrete Filterung mit FIR-Filter
-----------------------------------	---

Zwingende Voraussetzungen

Empfohlene Literatur	▪ Hamblen, Furman: Rapid Prototyping of Digital Systems, Kluwer Academic Publishing ▪ Wakerly: Digital Design: Principles and Practices, Prentice Hall ▪ D. Gajski: Embedded System Design, Springer Verlag New York ▪ U. Meyer-Baese: Digital Signal Processing with Field Programmable Gate Arrays
-----------------------------	--

Verwendung des Moduls in weiteren Studiengängen	▪ SOP in Bachelor Medientechnologie 2024 ▪ SOP in Bachelor Informatik und Systems-Engineering 2024
--	--

Besonderheiten und Hinweise

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.56 SWP - Softwarepraktikum

Modulkürzel	SWP_BaTIN2020
Modulbezeichnung	Softwarepraktikum
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	SWP - Softwarepraktikum
ECTS credits	6
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Hans Nissen/Professor Fakultät IME
Dozierende*r	Prof. Dr. Hans Nissen/Professor Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt unterschiedliche Kompetenzen, die für die team-orientierte Entwicklung komplexer Software-Systeme erforderlich sind:
der Entwurf von System-Komponenten gemäß Spezifikation und textuellen Anforderungen im Team (K.1, K.4, K.2, K.3, K.5, K.10, K.13),
die Implementierung von System-Komponenten gemäß Entwurf im Team (K.13, K.16, K.6, K.10),
die Prüfung einer implementierten Komponente auf Korrektheit (K.7),
die technische Dokumentation einer Implementierung (K.8, K.16),
die Integration von Komponenten eines Systems zusammen mit anderen Entwicklerteams (K.1, K.10, K.13, K.16),
die Prüfung eines integrierten System auf Korrektheit (K.7),
das zeitliche und inhaltliche Strukturieren und Organisieren eines eigenen Projekt (K.11, K.13),
das (teilweise) eigenständige Erlernen und Anwenden verschiedener typischer Werkzeuge in der Praxis (K.9, K.15).

Womit: Die Veranstaltung besteht aus einem einsemestrigen Projekt, welches die Studierenden in einem Team aus drei bis fünf Mitgliedern bearbeiten.
Der Schwerpunkt der Projektarbeit der Studierenden liegt dabei auf der Umsetzung von Anforderungen und Spezifikationen in einem lauffähigen Gesamtsystem.
Der Dozent gibt die Anforderungsspezifikation, die System-Spezifikation und den Grobentwurf des Gesamtsystems vor. Jedes Team entwirft, realisiert und testet eine Komponente des Gesamtsystems und integriert die eigenen Komponenten mit den Komponenten anderer Teams zu einem lauffähigen Gesamtsystem.
Jedes Team organisiert ihr Entwicklungsprojekt selbst. Vorgegeben durch den Dozenten werden die wesentlichen Meilensteine des Projekts, an denen die Ergebnisse mit den Teams besprochen und bewertet werden.

Wozu: In dem praxisnahen Projekt in dieser Veranstaltung sammeln die Studierenden praktische, realitätsnahe Erfahrungen in allen typischen Arbeitsfeldern eines Bachelorabsolventen:
die Studierenden entwickeln eine Teilkomponente eines spezifizierten Gesamtsystems selbständig in einem Teams (HF.1),
sie analysieren und bewerten die entwickelte Komponente bezüglich einer vorgegebenen Spezifikation und das Gesamtsystem bezüglich der Kundenanforderungen (HF.2),
sie organisieren ihr eigenes Entwicklungsprojekt zeitlich und (teilweise) inhaltlich selbst und übernehmen selbstverantwortlich die Strukturierung einer Komponente des Gesamtsystems (HF.3),
die Studierenden kommunizieren während des gesamten Projekts intensiv und über verschiedene Kanäle mit dem Dozenten als dem Auftraggeber und späterem Anwender des Projekts, den Mitgliedern des eigenen Teams und in der Phase der Integration auch mit den Mitgliedern mehrerer anderer Teams (HF.4).

Modulinhalte**Projekt**

Umgang mit semi-formalen Systemspezifikationen

Software-Erstellung im Team

eigenes Projekt zeitlich und inhaltlich strukturieren und organisieren

Umgang mit typischen Werkzeugen aus der Praxis

Entwicklungsumgebung

Versions-Management

Fehler-Management

Test-Werkzeug

Kooperations- und Kommunikations-Unterstützung

verschiedene Java-APIs

Datenbank

System-Komponente gemäß Spezifikation und textuellen Anforderungen entwerfen

System-Komponente gemäß Entwurf im Team implementieren

implementierte Komponente auf Korrektheit prüfen

Implementierung aus technischer Sicht dokumentieren

Komponenten eines Systems zusammen mit anderen Entwicklerteams integrieren

integriertes System auf Korrektheit prüfen

Lehr- und Lernmethoden	Projekt
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	180 Stunden
Präsenzzeit	12 Stunden $\pm$ 1 SWS
Selbststudium	168 Stunden
Empfohlene Voraussetzungen	▪ Modul PI1: sehr gute praktische Beherrschung der Programmiersprache Java ▪ Modul PP: Das Programmierpraktikum vermittelt bestimmte, spezifische softwarearchitektonisch Voraussetzungen, Stile und Idiome, auf denen das Softwarepraktikum aufbaut. ▪ Modul DB1: Grundlagen der Speicherung in einer Datenbank und der Anfrage von Daten ▪ sehr gute Programmierkenntnisse ▪ Kenntnissen in Software Engineering ▪ Kenntnisse in Datenbanken
Zwingende Voraussetzungen	▪ Modul SE: Spezifikation und Modellierung von Systemen und Software mit UML, Modularisierung in Java, einfache Entwurfsmuster, grundlegende Verfahren zum Prüfen von Software, verschiedene Architekturen von Systemen und Software, Grundbegriffe der Qualitätssicherung, Kenntnisse in Versionsverwaltung ▪ Modul PI2: Das Softwarepraktikum setzt Konzepte für das Programmieren im Großen voraus, die in PI2 vermittelt werden. Fehlen diese Voraussetzungen bei einzelnen Team-Mitgliedern, ist der Team-Erfolg im Softwarepraktikum gefährdet. ▪ Projekt erfordert Anwesenheit im Umfang von: 7 Termine
Empfohlene Literatur	▪ I. Sommerville: Software Engineering, Addison-Wesley, 2007. ▪ H. Balzert: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, Spektrum Akademischer Verlag, 3. Auflage, 2012. ▪ B. Brügge, A.H. Dutoit: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java, Pearson Studium, 2006. ▪ H. Balzert: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, Spektrum Akademischer Verlag, 3. Auflage, 2009.

Verwendung des Moduls in weiteren Studiengängen	SWP in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise	
------------------------------------	--

Letzte Aktualisierung	19.7.2025, 14:32:16
------------------------------	---------------------

6.57 SYP - Systementwicklungs-Projekt

Modulkürzel	SYP_BaTIN2020
Modulbezeichnung	Systementwicklungs-Projekt
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	SYP - Systementwicklungs-Projekt
ECTS credits	7
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	5
Häufigkeit des Angebots	jedes Wintersemester
Modul-Verantwortliche*r	Prof. Dr. René Wörzberger/Professor Fakultät IME
Dozierende*r	Prof. Dr. René Wörzberger/Professor Fakultät IME
Learning Outcome(s)	
(WAS) Studierende entwerfen und entwickeln im Rahmen eines eigenen Software-Entwicklungsprojekts im Team ein Software-System (WOMIT) indem sie das Projekt geeignet planen, Anforderungen systematisch erheben, das System nach geeigneten Qualitätskriterien entwerfen, zunächst prototypische implementieren und präsentieren, durch geeignete Maßnahmen (Tests etc.) qualitätssichern und in benutzbarer und dokumentierter Form an ihren Auftraggeber übergeben (WOZU) damit sie im späteren Beruf als Mitglied eines Entwicklungsteams wirksam mitarbeiten können.	
Modulinhalte	
Praktikum	
- SW-Erstellung im Team - angemessene Kommunikation mit einem Kunden - Bearbeitung des gesamten Software-Lebenszyklus - eigenes Projekt zeitlich und inhaltlich strukturieren und organisieren - Präsentationen erstellen und halten - eigenen Zeitplan für das Projekt erstellen - System-Anforderungen vom Kunden ermitteln und dokumentieren - System gemäß Anforderungen im Team spezifizieren und modellieren - System gemäß funktionaler Spezifikation und Qualitätsanforderungen im Team entwerfen - System gemäß Entwurf im Team implementieren - technische Details des realisierten Systems aussagekräftig darstellen - implementiertes System auf Korrektheit prüfen - Benutzung des realisierten Systems dokumentieren - Verteidigung der eigenen Lösungen	
Lehr- und Lernmethoden	Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	210 Stunden
Präsenzzeit	12 Stunden $\pm$ 1 SWS
Selbststudium	198 Stunden

Empfohlene Voraussetzungen	▪ Modul PI1: Die zwingende Voraussetzung PI2 baut auf PI1 auf. ▪ Modul SWP: Das Arbeiten im Team sollte zunächst bei der enger gefassten Aufgabenstellung des SWP eingeübt werden.
Zwingende Voraussetzungen	▪ Modul PI2: Programmierkenntnisse auf dem Niveau von PI2 sind unabdingbar für einen ausreichenden Beitrag im jeweiligen SYP-Team ▪ Modul SE: In SYP sollen die Kenntnisse in sämtlichen Arbeitsbereichen des Software-Engineerings (SE) angewendet werden, die zuvor in SE vermittelt wurden. Es reicht für die Teilnahme in SYP, wenn das Praktikum in SE bestanden wurde. ▪ Praktikum erfordert Anwesenheit im Umfang von: 5 Termine
Empfohlene Literatur	▪ keine
Verwendung des Moduls in weiteren Studiengängen	▪ SYP in Bachelor Informatik und Systems-Engineering 2024 ▪ PLSYP in Master Technische Informatik 2020 ▪ PLSYP in Master Informatik und Systems-Engineering 2024
Perma-Links zur Organisation	<u>Ilu-Kurs</u>
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.58 UT - Übertragungstechnik

Modulkürzel	UT_BaTIN2020
Modulbezeichnung	Übertragungstechnik
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	UT - Übertragungstechnik
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Uwe Dettmar/Professor Fakultät IME
Dozierende*r	Prof. Dr. Uwe Dettmar/Professor Fakultät IME

Learning Outcome(s)

Was? Die Übertragung von in binärer Form vorliegender Daten über gestörte Kanäle durch Modulation inklusive des Entwurfs von Modulator und Demodulator
Womit? Unter Anwendung von Verfahren und Algorithmen der digitalen Übertragungstechnik
Wozu? Zur Realisierung einer an die Eigenschaften des Kanals angepassten zuverlässigen Datenübertragung in kommunikationstechnischen Systemen.

Modulinhalte**Vorlesung / Übungen**

Vorlesung und Übungen werden in einer Lehrveranstaltung kombiniert. Nach der Vorstellung von neuem Lernstoff durch den Dozenten in Form von kurzen Blöcken wird dieser direkt von den Studierenden durch kurze Matlab- und Python-Übungen angewendet und vertieft. Längere Übungsaufgaben werden bereits zu Hause vorbereitet und die verschiedenen Lösungsvorschläge in der Präsenzveranstaltung besprochen.

Über ein Lernportal werden elektronische Minitests zum aktuell behandelten Stoff als weitere Lernressource angeboten.

Inhalte:

- Geschichte der Nachrichtentechnik
- Modelle und Inhalte der Übertragungstechnik
- Grundbegriffe wie Bandbreite, Datenrate, Baudrate etc.
- Signale, Systeme und Modulationsverfahren
- Mehrträgerverfahren
- Übertragungskanäle und Elemente digitaler Übertragungssysteme
- Entscheidungstheorie
- Link Budget Berechnung

Die Studierenden lernen die o.g. Themen in der Vorlesung kennen, erwerben Grundwissen und vertiefen dieses durch Selbststudium mit Hilfe von Literatur, YouTube Videos und anderen Netzressourcen (selbstständige Informationsbeschaffung), sowie in Lerngruppen (Teamwork).

Durch kleinere Übungsaufgaben und Programme wird in der Präsenzveranstaltung bereits ein aktiver Umgang mit den vorgestellten Verfahren trainiert. Umfangreichere Rechenaufgaben werden am Ende der Veranstaltung behandelt und die Lösungswege diskutiert, um dadurch den Studierenden relevante Problemstellungen vorzustellen und ihre Fähigkeit zur Lösungsfindung zu entwickeln.

Die Studierenden lernen darüber hinaus:

- nachrichtentechnische System zu analysieren und deren Performanz zu ermitteln bzw. abzuschätzen.
- Verfahren der Übertragungstechnik zu vergleichen und zu bewerten
- Kenntnisse auf technische Problemstellungen anzuwenden

Praktikum

Bearbeitung von geeigneten Praktikumsaufgaben aus dem Bereich der Übertragungstechnik in Form von Jupyter Notebooks und Python Programmen. Die Studierenden verwenden dabei teils fertige oder vorhandene Programme für Simulationen. Sie notieren die Ergebnisse, erzeugen graphische Darstellungen und diskutieren die Ergebnisse.

Matlab mit der Communications Toolbox wird für Simulationsaufgaben verwendet, deren zeitlicher Aufwand für eine Eigenentwicklung zu groß ist.

- Die Studierenden schulen ihre Fähigkeiten zur Lösung technischer Probleme mit Hilfe von Computerprogrammen.
- Sie analysieren und simulieren nachrichtentechnische Systeme und bewerten deren Eigenschaften.
- Sie schulen ihre Selbstorganisation und ihr problemorientiertes Denken und Handeln.
- Sie trainieren das Lösen von Aufgaben im Team und ihre kommunikativen Fähigkeiten.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul MA1: Elementare Funktionen, Differentialrechnung, Lineare Gleichungssysteme ▪ Modul MA2: Komplexe Rechnung, Integralrechnung, Lineare Algebra ▪ Modul EG: Grundbegriffe, elektrische und magnetische Feldgrößen, Komplexe Wechselstromrechnung ▪ Modul SIG: Signale, Impulsantwort, Faltung, Fourier Transformation und Spektren ▪ Die Studierenden sollten Grundkenntnisse in den Gebieten Lineare Algebra und Stochastik und zusätzlich Programmierkenntnisse mitbringen, die es Ihnen ermöglichen, einfache Programme in einer höheren Programmiersprache zu schreiben. In der Vorlesung werden Matlab/Octave und Python verwendet. Die Studierenden sollten physikalische Größen und Einheiten verwenden können und Grundkenntnisse der komplexen Wechselstromrechnung besitzen. Außerdem sollten sie Grundfertigkeiten aus der Signaltheorie und die Fouriertransformation beherrschen.
Zwingende Voraussetzungen	▪ Praktikum erfordert Anwesenheit im Umfang von: 4 Praktikumstermine ▪ Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	▪ BOSSERT, M. : Einführung in die Nachrichtentechnik. Oldenbourg Verlag, 2012. ▪ MEYER, M. : Kommunikationstechnik. 4. Vieweg und Teubner, 2019. ▪ JOHNSON, SETHARES, KLEIN: Software Receiver Design, Cambridge 2011 ▪ PROAKIS, J. G. ; SALEHI, M. : Digital Communications. 5. McGraw-Hill, 2008.
Verwendung des Moduls in weiteren Studiengängen	▪ UT in Bachelor Elektrotechnik 2020 ▪ UT in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ UT in Bachelor Informatik und Systems-Engineering 2024
Besonderheiten und Hinweise	
Letzte Aktualisierung	19.7.2025, 14:32:16

6.59 VDS - Verteilte Datenverarbeitungssysteme

Modulkürzel	VDS_BaTIN2020
Modulbezeichnung	Verteilte Datenverarbeitungssysteme
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	VDS - Verteilte Datenverarbeitungssysteme
ECTS credits	5
Sprache	deutsch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Semester
Modul-Verantwortliche*r	Prof. Dr. Andreas Behrend/Professor Fakultät IME
Dozierende*r	Prof. Dr. Andreas Behrend/Professor Fakultät IME

Learning Outcome(s)

Was: Im Rahmen der Vorlesung sollen prinzipielle Big Data Konzepte vermittelt werden. Das beinhaltet Fault Tolerance in verteilten Systemen, verschiedene Fehlerbehandlungsmechanismen (z.B. Replikation, Wiederherstellung) sowie Bloom Filter und N-Grams als Techniken zur Optimierung von Datenabfragen. Zudem werden Konzepte wie die Jaccard Similarity sowie Min/LS Hashing vorgestellt, um große Datenmengen effizient vergleichen und analysieren zu können sowie die grundlegenden Prinzipien von NoSQL-Systemen, einschließlich Replikation, CAP-Theorem, Multiversion- und BASE-Konsistenz, Consistent Hashing, Casual Consistency, Time-to-Live (TTL), Lamport- und Vector Clocks vermittelt.

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. In den Übungen und insbesondere im Praktikum erarbeiten die Studierenden Softwarelösungen auf Basis von NoSQL-Systemen in Kleingruppen. Im Praktikum analysieren die Studierenden dafür Problemstellungen zur Datenspeicherung und -analyse (K1, K2, K4), programmieren Lösungen auf der Grundlage anerkannter Konzepte und Methoden (K3) mit Hilfe von aktuellen Systemen und Werkzeugen (K6, K9) und evaluieren die Softwarelösung (K7). Dazu verwenden sie verfügbare Dokumentationen (K8, K15) und passen die Softwaresysteme bzw. Werkzeuge entsprechend an (K10). Zum Abschluss des Praktikums müssen die Studierenden ihre Lösungen vorstellen und verteidigen (K8, K13, K16).

Wozu: In komplexen Softwaresystemen muss die Verarbeitung großer, heterogener Datenmengen oft nebenläufig und verteilt erfolgen. Kenntnisse über die Grundlagen und die Programmierung verteilter Datenverarbeitungssysteme sind somit essentiell für die Erstellung moderner Software (HF1). Durch ihre praktische Programmierarbeit erwerben die Studierenden Erfahrungen, die wichtig sind für die Erfassung von Anforderungen, die Entwicklung von Konzepten zur technischen Lösung und zu ihrer Bewertung (HF2) sowie zur Organisation bzw. zum Betrieb von Systemen, die nebenläufig und verteilt arbeiten (HF3). Die Durchführung im Team und mit dem Dozenten als "Auftraggeber" stärkt die Interaktionsfähigkeit der Studierenden (HF 4).

Durch die Auseinandersetzung mit diesen Themen entwickeln die Studierenden das nötige Wissen und die Fähigkeiten, um verteilte Datenverarbeitungssysteme zu entwerfen, zu implementieren und zu optimieren, die für moderne Big Data-Anwendungen erforderlich sind.

Modulinhalte**Vorlesung / Übungen**

Big Data Konzepte wie etwa OLTP, OLAP, RTAP, Distributed Computing und NoSQL-Systeme
 Konzepte zur nebenläufigen Berechnungen von großen Datenmengen (Map Reduce, Hadoop, Pig, Hive, Cluster Computing)
 Konzepte zur nebenläufige Datenanalyse (Bloom Filter, N-Grams, Jaccard Similarity, Min/LS Hashing)
 Eigenschaften von NoSQL-Systemen (Replication, CAP-Theorem, Lamport/Vector Clocks, Multiversion, BASE, Quorums)
 NoSQL-Systeme:
 Key Value and Column-Family (Consistent Hashing, Casual Consistency, TTL, Filtering)
 Document Stores and Graph DB (JSON, Replica Management, Elections, Cypher)
 Datenstromsysteme (Probabilistic Counting, Sliding Windows, Push vs. Pull, CQL)

Die Studierenden können NoSQL-Systeme einordnen und deren Eignung für vorgegebene Anforderungen bewerten.
 Sie können die unterschiedlichen Architekturen für verteilte Datenverarbeitungssysteme erkennen und deren Vor- bzw. Nachteile analysieren. Sie sind darüber hinaus in der Lage, große Systeme zur verteilten Datenverarbeitung einzurichten und nutzerspezifisch zu konfigurieren. Zudem können die Studierenden beispielhafte Analysen mit den Anfragesprachen derartiger Systeme implementieren.

Praktikum

Ziel des Praktikums ist es, den Studierenden die Fähigkeiten zu vermitteln, Graphdatenbanken und Datenstromsysteme effektiv zu nutzen, um komplexe, datenintensive Probleme zu lösen und praxisorientierte Anwendungen zu entwickeln. Die Studierenden lernen, wie man Daten mit dem Graphdatenbanksystem Neo4j modelliert und wie man mit Cypher komplexe Abfragen erstellt, um Beziehungen zwischen verschiedenen Entitäten zu analysieren. Zudem trainieren sie, wie man Neo4j effizient nutzt, indem sie Abfragen optimieren, mit großen Datenmengen umgehen und mögliche Skalierungsstrategien verstehen.

Daneben lernen die Studierenden die Konzepte von Echtzeit-Datenströmen, Stream Processing und Event-driven Architectures kennen und verstehen, wie Daten kontinuierlich verarbeitet und analysiert werden. Sie entwickeln und implementieren Datenstrom-Pipelines, die Daten in Echtzeit sammeln, verarbeiten und weiterleiten. Sie lernen, wie man verschiedene Quellen und Senken (z. B. Datenbanken, Messaging-Systeme) integriert. In beispielhaften Anwendungsszenarien verwenden die Studierenden Datenstromsysteme (z.B. Apache Flink, Kafka, etc.), um Daten in Echtzeit zu analysieren, z. B. zur Echtzeit-Überwachung oder zur Anomalieerkennung.

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	45 Stunden $\pm$ 4 SWS
Selbststudium	105 Stunden
Empfohlene Voraussetzungen	▪ Modul PI1: Sicherer Umgang mit einer Programmiersprache. ▪ Modul PI2: Sicherer Umgang mit einer Programmiersprache. ▪ Modul DB1: Datenanalyse mittels SQL sowie Architektur von Datenverarbeitungssystemen ▪ Veranstaltung Datenbanken I ▪ Veranstaltung Datenbanken II
Zwingende Voraussetzungen	Teilnahme an abschließender Prüfung nur nach erfolgreicher Teilnahme an Praktikum
Empfohlene Literatur	▪ Martin Kleppmann: "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems" ▪ Pramod J. Sadalage und Martin Fowler: "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" ▪ Lena Wiese: "NoSQL-Datenbanken: Konzepte – Technologien – Anwendungen"
Verwendung des Moduls in weiteren Studiengängen	VDS in Bachelor Informatik und Systems-Engineering 2024

**Besonderheiten und
Hinweise**

Letzte Aktualisierung 19.7.2025, 14:32:16

6.60 VMA - Programmierung verteilter und mobiler Anwendungen

Modulkürzel	VMA_BaTIN2020
Modulbezeichnung	Programmierung verteilter und mobiler Anwendungen
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	MPR - Mobilgeräteprogrammierung
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr. Carsten Vogt/Professor Fakultät IME
Dozierende*r	▪ Prof. Dr. Carsten Vogt/Professor Fakultät IME ▪ Marcel Henk/ wissenschaftlicher Mitarbeiter Fakultät IME

Learning Outcome(s)

Was: Das Modul vermittelt Kompetenzen zur Programmierung von Mobilgeräten, insbesondere von Smartphones. In praktischer Arbeit analysieren die Studierenden Problemstellungen (K2), implementieren Lösungen mit Hilfe von Standardwerkzeugen (K6, K9) und prüfen sie (K7). Sie recherchieren dazu in Online-Dokumentationen (K8, K15) und passen vorhandene Software an (K10). Darüber hinaus befähigt das Modul die Studierenden, die Folgen bei der Programmierung und beim Einsatz von Mobilgeräten einzuschätzen (K14).

Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs-/Übungsteil und betreut darauf aufbauend ein Praktikum. In den Übungen und insbesondere im Praktikum arbeiten die Studierenden in Kleingruppen und verteidigen ihre Lösungen (K8, K13, K16).

Wozu: Mobilgeräte spielen im privaten und professionellen Umfeld eine zentrale Rolle und somit auch Kenntnisse, sie zu programmieren und in verteilte Systeme zu integrieren (HF1). Durch ihre praktische Programmierarbeit erwerben die Studierenden zudem weitere Erfahrungen, die wichtig sind für die Erfassung von Anforderungen, die Entwicklung von Konzepten zur technischen Lösung und zu ihrer Bewertung (HF2). Die Durchführung im Team mit dem Dozenten als "Auftraggeber" stärkt die Interaktionsfähigkeit der Studierenden (HF 4).

Modulinhalte**Vorlesung / Übungen**

grundlegende Begriffe und Techniken
 Eigenschaften von Mobilgeräten im Vergleich zu traditionellen Computern
 Übersicht über aktuelle Mobilgeräte-Betriebssysteme und -Programmiersplattformen
 grundlegende Vorgehensweise bei der Programmierung eines Mobilgeräts (Programmerstellung, -emulation und -installation)
 Programmierung von Mobilgeräten am Beispiel eines oder mehrerer aktueller Systeme (Randbemerkung: Dieser (Haupt-)Teil der Lehrveranstaltung wird jeweils den aktuellen technischen Gegebenheiten und der aktuellen Marktsituation angepasst. An dieser Stelle werden daher die voraussichtlich zu behandelnden Themengebiete nur grob genannt.)
 Komponenten einer Mobilgeräte-Applikation
 graphische Oberflächen
 Datenhaltung
 Nebenläufigkeit
 Datenkommunikation, insbes. Internetzugriff
 ortsabhängige Dienste
 Sicherheit
 Umgang mit Softwareentwicklungsumgebungen für Mobilgeräte
 Programmtechnische Realisierung von Mobilgeräte-Applikationen mittlerer Komplexität
 Bewertung der Risiken bei der Programmierung und beim Gebrauch von Mobilgeräten

Praktikum

Programmierung von Smartphones am Beispiel eines aktuellen Systems - Details wie unter "Vorlesung/Übung" angegeben
 Umgang mit Softwareentwicklungsumgebungen für Smartphones
 Realisierung von Smartphone-Applikationen mittlerer Komplexität in kleinen Teams

Lehr- und Lernmethoden	▪ Vorlesung / Übungen ▪ Praktikum
-------------------------------	--

Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
--------------------------------------	-----------------------

Workload	150 Stunden
-----------------	-------------

Präsenzzeit	45 Stunden $\pm$ 4 SWS
--------------------	------------------------

Selbststudium	105 Stunden
----------------------	-------------

Empfohlene Voraussetzungen	▪ Modul PI1: Sicherer Umgang mit einer objektorientierten Programmiersprache. ▪ Modul PI2: Sicherer Umgang mit einer objektorientierten Programmiersprache. ▪ Modul BVS1: Struktur und Funktionalität von Betriebssystemen. Grundkenntnisse in der nebenläufigen Programmierung (Threading) und in der Netzwerkprogrammierung (Sockets). ▪ Modul DB1: Grundkenntnisse in relationalen Datenbanken, auch Programmierung damit. ▪ Modul NP: Grundkenntnisse in Internet-Protokollen. ▪ Objektorientierte Programmierung (idealerweise Java) Struktur und Funktionalität von Betriebssystemen Programmierung mit Nebenläufigkeit / Threading und Sockets Kommunikationsprotokolle für Datennetze Relationale Datenbanken
-----------------------------------	--

Zwingende Voraussetzungen	Praktikum erfordert Anwesenheit im Umfang von: 1 Testattermin
----------------------------------	---

Empfohlene Literatur	▪ Aufgrund des sehr dynamischen Fachgebiets können Literaturangaben immer nur jeweils unmittelbar zu Veranstaltungsbeginn gemacht werden. Aktuell: http://developer.android.com; Künneth, Android 8 - Das Praxisbuch für Entwickler, Rheinwerk 2018
-----------------------------	--

Verwendung des Moduls in weiteren Studiengängen	▪ VMA in Bachelor Elektrotechnik 2020 ▪ MPR in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ MPR in Bachelor Informatik und Systems-Engineering 2024
--	---

Besonderheiten und Hinweise

Letzte Aktualisierung 19.7.2025, 14:32:16

6.61 XIB - Fachübergreifende Kompetenzen und Soft-Skills

Modulkürzel	XIB_BaTIN2020
Modulbezeichnung	Fachübergreifende Kompetenzen und Soft-Skills
Art des Moduls	Pflichtmodul
Anerkannte Lehrveranstaltung	XGA - Gremienarbeit
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	5
Häufigkeit des Angebots	jedes Semester
Modul-Verantwortliche*r	Studiengangsleiter(in) Bachelor Technische Informatik / Informatik und Systems-Engineering
Dozierende*r	
Learning Outcome(s)	Die Studierenden lernen, über die fachbezogenen Grenzen ihres Studiums hinweg zu schauen. Sie sind in der Lage, internationale, inter-/transdisziplinäre und/oder interkulturelle Aspekte ihres zukünftigen Berufs zu erkennen, einzuordnen, ihr Verhalten darauf einzustellen und auch in fremdem Kontext sicher zu agieren. Das konkrete Lehrangebot wird in der Regel erst kurzfristig zu Beginn des jeweiligen Semesters festgelegt. Es kann unterschiedlichste Themen behandeln; eine Zusammenarbeit mit anderen Fakultäten oder Instituten ist vorgesehen. Je nach konkret gewähltem Lehrangebot werden die u.a. Kompetenzen unterschiedlich intensiv vermittelt. Das Modul kann auch durch Teilnahme an mehreren verschiedenen kleineren Lehrveranstaltungen erfüllt werden, sofern diese zu den Modulzielen beitragen und die erforderlichen ECTS-Punkte in Summe erreicht sind.
Modulinhalte	
Projekt	
Lehr- und Lernmethoden	Projekt
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	12 Stunden $\pm$ 1 SWS
Selbststudium	138 Stunden
Empfohlene Voraussetzungen	
Zwingende Voraussetzungen	
Empfohlene Literatur	
Verwendung des Moduls in weiteren Studiengängen	▪ XIB in Bachelor Elektrotechnik 2020 ▪ XGA in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ XGA in Bachelor Informatik und Systems-Engineering 2024 ▪ XGA in Master Informatik und Systems-Engineering 2024

Besonderheiten und Hinweise	Anerkennbar ist die Mitarbeit in Berufungskommissionen als studentisches Mitglied. Die Anzahl der anerkannten ECTS-Punkte richtet sich nach der Anzahl der nachgewiesenen Stunden in der Gremientätigkeit. Es wird 1ECTS-Punkt pro 25 Stunden Gremienarbeit angerechnet. Der/die Vorsitzende der Berufungskommission vergibt die ECTS und bescheinigt diese. Es wird erwartet, dass der/die Studierende sich aktiv in die Arbeit einbringt.
Letzte Aktualisierung	19.7.2025, 14:32:16

6.62 XIB2 - Fachübergreifende Kompetenzen und Soft-Skills

Modulkürzel	XIB2_BaTIN2020
Modulbezeichnung	Fachübergreifende Kompetenzen und Soft-Skills
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	XGA - Gremienarbeit
ECTS credits	5
Sprache	deutsch, englisch bei Bedarf
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Semester
Modul-Verantwortliche*r	Studiengangsleiter(in) Bachelor Technische Informatik / Informatik und Systems-Engineering
Dozierende*r	
Learning Outcome(s)	Die Studierenden lernen, über die fachbezogenen Grenzen ihres Studiums hinweg zu schauen. Sie sind in der Lage, internationale, inter-/transdisziplinäre und/oder interkulturelle Aspekte ihres zukünftigen Berufs zu erkennen, einzuordnen, ihr Verhalten darauf einzustellen und auch in fremdem Kontext sicher zu agieren. Das konkrete Lehrangebot wird in der Regel erst kurzfristig zu Beginn des jeweiligen Semesters festgelegt. Es kann unterschiedlichste Themen behandeln; eine Zusammenarbeit mit anderen Fakultäten oder Instituten ist vorgesehen. Je nach konkret gewähltem Lehrangebot werden die u.a. Kompetenzen unterschiedlich intensiv vermittelt. Das Modul kann auch durch Teilnahme an mehreren verschiedenen kleineren Lehrveranstaltungen erfüllt werden, sofern diese zu den Modulzielen beitragen und die erforderlichen ECTS-Punkte in Summe erreicht sind.
Modulinhalte	
Projekt	
Lehr- und Lernmethoden	Projekt
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	12 Stunden $\pm$ 1 SWS
Selbststudium	138 Stunden
Empfohlene Voraussetzungen	
Zwingende Voraussetzungen	
Empfohlene Literatur	
Verwendung des Moduls in weiteren Studiengängen	▪ XIB in Bachelor Elektrotechnik 2020 ▪ XGA in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ XGA in Bachelor Informatik und Systems-Engineering 2024 ▪ XGA in Master Informatik und Systems-Engineering 2024

Besonderheiten und Hinweise	Anerkennbar ist die Mitarbeit in Berufungskommissionen als studentisches Mitglied. Die Anzahl der anerkannten ECTS-Punkte richtet sich nach der Anzahl der nachgewiesenen Stunden in der Gremientätigkeit. Es wird 1ECTS-Punkt pro 25 Stunden Gremienarbeit angerechnet. Der/die Vorsitzende der Berufungskommission vergibt die ECTS und bescheinigt diese. Es wird erwartet, dass der/die Studierende sich aktiv in die Arbeit einbringt.
Letzte Aktualisierung	19.7.2025, 14:32:16

6.63 XPSS - Praxisorientierte Summer School

Modulkürzel	XPSS_BaTIN2020
Modulbezeichnung	Praxisorientierte Summer School
Art des Moduls	Wahlpflichtmodul
Anerkannte Lehrveranstaltung	XPSS - Praxisorientierte Summer School
ECTS credits	5
Sprache	englisch
Dauer des Moduls	1 Semester
Empfohlenes Studiensemester	4-6
Häufigkeit des Angebots	jedes Sommersemester
Modul-Verantwortliche*r	Prof. Dr.-Ing. Ulrich Reiter/Professor Fakultät IME
Dozierende*r	
Learning Outcome(s)	In dem Modul lernen die Studierenden die Zusammenarbeit in kleinen internationalen Teams. Dazu analysieren sie Problemstellungen und erstellen geeignete kreative Lösungskonzepte, die in Form von Vorträgen präsentiert werden. Sie sammeln praktische Erfahrung in der Realisierung kleinerer Projekte und der Präsentation der erreichten Ergebnisse. Durch die Arbeit in internationalen Teams vertiefen die Studierenden ihre interkulturellen Fähigkeiten. Als Ergebnis des Moduls sind die Teilnehmer und Teilnehmerinnen in der Lage, fachspezifische Aufgabenstellungen zu analysieren, Lösungskonzepte zu entwickeln und technische Systeme in einer internationalen Umgebung zu erstellen.
Modulinhalte	
Projekt	Arbeiten in kleinen Teams, Selbstorganisation, Projektplanung, Projektrealisierung, Präsentation
Lehr- und Lernmethoden	Projekt
Prüfungsformen mit Gewichtung	siehe Prüfungsordnung
Workload	150 Stunden
Präsenzzeit	12 Stunden $\pm$ 1 SWS
Selbststudium	138 Stunden
Empfohlene Voraussetzungen	▪ Modul EG: Anwenden einfacher elektrischer Schaltungen, ▪ Modul PI1: Erstellen von funktionsfähigen Programmen ▪ Modul MA1: Anwenden grundlegender mathematischer Kenntnisse ▪ Modul MA2: Anwenden fortgeschrittener mathematischer Kenntnisse ▪ Gutes Verständnis in der Programmierung von Mikrocontrollern. Kenntnisse in der Funktion von elektronischen Bauelementen und Komponenten. Praktische Fähigkeiten in der Realisierung von Schaltungen.
Zwingende Voraussetzungen	
Empfohlene Literatur	
Verwendung des Moduls in weiteren Studiengängen	▪ XPSS in Bachelor Elektrotechnik 2020 ▪ XPSS in Bachelor Elektrotechnik und Informationstechnik 2024 ▪ XIM in Master Technische Informatik 2020

Besonderheiten und Hinweise	Die Lehrveranstaltung wird in Abstimmung mit externen Hochschulen angeboten. Ein fester Zeitraum kann nicht angegeben werden. Die Lehrveranstaltung wird ausreichend früh angekündigt.
Letzte Aktualisierung	19.7.2025, 14:32:16

7. Wahlbereiche

Im Folgenden wird dargestellt, welche Module in einem bestimmten Wahlbereich gewählt werden können. Für alle Wahlbereiche gelten folgende Hinweise und Regularien:

- **Bei der Wahl von Modulen aus Wahlbereichen gelten zusätzlich die Bedingungen, die im Abschnitt Studienschwerpunkte formuliert sind.**
- In welchem Semester Wahlpflichtmodule eines Wahlbereichs typischerweise belegt werden können, kann den Studienverlaufsplänen entnommen werden.
- Module werden in der Regel nur entweder im Sommer- oder Wintersemester angeboten. Das heißt, dass eine eventuell erforderliche begleitende Prüfung nur im Sommer- oder Wintersemester abgelegt werden kann. Die summarischen Prüfungen werden bei Modulen der Fakultät 07 für Medien-, Informations- und Elektrotechnik in der Regel in der Prüfungszeit nach jedem Semester angeboten.
- Ein absolviertes Modul wird für maximal einen Wahlbereich anerkannt, auch wenn es in mehreren Wahlbereichen aufgelistet ist.
- Bei manchen Modulen gibt es eine Aufnahmebegrenzung. Näheres hierzu ist in den Bekanntmachungen zu den Aufnahmebegrenzungen zu finden.
- Die Anmeldung an und die Aufnahme in fakultätsexterne Module unterliegen Fristen und anderen Bedingungen der anbietenden Fakultät oder Hochschule. Eine Aufnahme kann nicht garantiert werden. Studierende müssen sich frühzeitig bei der jeweiligen externen Lehrperson informieren, ob Sie an einem externen Modul teilnehmen dürfen und was für eine Anmeldung und Teilnahme zu beachten ist.
- Auf Antrag kann der Wahlbereich um weitere passende Module ergänzt werden. Ein solcher Antrag ist bis spätestens vier Monate vor einer geplanten Teilnahme an einem zu ergänzenden Modul formlos an die Studiengangsleitung zu richten. Über die Annahme des Antrags befindet der Prüfungsausschuss im Benehmen mit der Studiengangsleitung und fachlich geeigneten Lehrpersonen. Eine anzuerkennende Studienleistung
 - muss sich in das intendierte AbsolventInnen-Profil des Studiengangs fügen und zu dessen Erreichung beitragen,
 - muss lernergebnisorientiert sein und darf nicht allein der Wissensvermittlung dienen,
 - muss mindestens dem Qualifikationsniveau eines Bachelorstudiengangs entsprechen,
 - muss einen vor dem Hintergrund des vorgesehenen Studienverlaufs sinnvollen Kompetenzzuwachs darstellen,
 - muss durch eine Prüfungsleistung abgeschlossen worden sein und
 - darf hinsichtlich ihrer Inhalte und Learning-Outcomes nicht mit bereits erfüllten Studienleistungen identisch sein.
- Im Folgenden sind Module nicht aufgeführt,
 - die in Vergangenheit lediglich im Rahmen individueller Anerkennungsverfahren für einen Wahlbereich anerkannt wurden oder
 - die in Vergangenheit lediglich im Rahmen eines Auslandsaufenthaltes und damit verbundenem, individuellem Learning-Agreements für einen Wahlbereich anerkannt wurden.

Auslandsaufenthalte

- Studierende, die einen Auslandsaufenthalt in ihr Studium integriert haben und dabei Studienleistungen an einer ausländischen Hochschule erbracht haben, können sich diese auf Antrag und mit Zustimmung des Prüfungsausschusses anerkennen lassen.
- Vor Antritt des Auslandsaufenthaltes ist mit dem Anerkennungsbeauftragten der Fakultät ein Learning-Agreement abzuschließen. Es wird dabei insbesondere vereinbart, für welche Pflichtmodule oder Wahlbereiche die im Ausland erbrachten Studienleistungen anerkannt werden.

7.1 AUS - Auslandssemester

Hier werden an einer ausländischen Hochschule erbrachte Leistungen nach vorheriger Absprache anerkannt, wenn ihr Umfang dem eines Semesters entspricht. Das konkrete Lehrangebot richtet sich nach der ausländischen Hochschule.

7.2 WM - Wahlbereich Informatik und Systems-Engineering

In diesem Wahlbereich können Module der Fakultät 07 für Informations-, Medien- und Elektrotechnik mit Bezug zur Technischen Informatik gewählt werden.

Aus diesem Wahlbereich müssen Module im Umfang von mindestens 35 ECTS-Kreditpunkten belegt werden.

Dieser Wahlbereich umfasst insbesondere alle Module aus folgenden anderen Bereichen:

- Studienschwerpunkt ES - Eingebettete Systeme
- Studienschwerpunkt IOT - Internet of Things
- Studienschwerpunkt NVS - Netze und Verteilte Systeme
- Studienschwerpunkt SMS - Smart Systems
- Studienschwerpunkt SOS - Software-Systeme

Module, die aus diesen anderen Bereichen stammen, sind im Folgenden normalgedruckt, originäre Module dieses Wahlbereichs sind fettgedruckt.

Module der Fakultät:

Modulkürzel	Modulbezeichnung	ECTS	enthalten in Studienschwerpunkt
ASN	Angewandte Statistik und Numerik	5	
AT	Antennentechnik	5	
ATS	Autonome Systeme	5	SOS ES SMS
BV	Bildverarbeitung	5	
CA	Computeranimation	5	SMS
CG	Computergrafik	5	SMS
DB2	Datenbanken 2	5	SOS
DM	Data Mining	5	SMS IOT
EKS	Entwicklung komplexer Software-Systeme	5	NVS SOS
EL	Elektronik	5	ES
ES	Eingebettete Systeme	5	ES
ESP	Eingebettete Systeme - Projekt	5	ES
FIT	Funksysteme für IoT	5	IoT
HF	Hochfrequenztechnik	5	
IAK	Ingenieurakustik	5	
IOT	IoT Protokolle und Anwendungen	5	NVS IoT
KOAK	Kommunikationsakustik	5	
ML	Maschinelles Lernen	5	SMS
MLO	Machine Learning Operations	5	
MPR	Mobilgeräteprogrammierung	5	
MT	Messtechnik	5	
NSA	Netzsicherheit und Automation	5	NVS
PPRA	Parallelprogrammierung und Rechnerarchitekturen	5	
QKC	Quellen- und Kanalcodierung	5	IoT
SM	Software Management	5	NVS SOS
SMO	Smart Mobility Components	5	IoT
SMP	Signalverarbeitung mit Matlab/Python und µC	5	IoT

Modulkürzel	Modulbezeichnung	ECTS	enthalten in	
			Studienschwerpunkt	
SOP	Systems on Programmable Chips	5	ES	IOT
UT	Übertragungstechnik	5		
VDS	Verteilte Datenverarbeitungssysteme	5		SMS
VMA	Programmierung verteilter und mobiler Anwendungen	5	NVS	SOS

7.3 XIB1 - Fachübergreifende Kompetenzen und Soft Skills 1

Die Module zu unterschiedlichen Themen werden in Zusammenarbeit mit anderen Fakultäten oder Institutionen angeboten. Das konkrete Lehrangebot wird in der Regel erst kurzfristig zu Beginn des jeweiligen Semesters festgelegt.

Aus diesem Wahlbereich müssen Module im Umfang von mindestens 5 ECTS-Kreditpunkten belegt werden.

Module der Fakultät:

Modulkürzel	Modulbezeichnung	ECTS	enthalten in Studienschwerpunkt
PHTB	Philosophische Handlungstheorie Bachelor	5	
XIB	Fachübergreifende Kompetenzen und Soft-Skills	5	
XPSS	Praxisorientierte Summer School	5	

8. Studienschwerpunkte

Im Folgenden wird dargestellt, welche Studienschwerpunkte in diesem Studiengang definiert sind (vgl. auch §24 der Prüfungsordnung). Für alle Studienschwerpunkte gelten folgende Hinweise und Regularien:

- Ein Studienschwerpunkt gilt als erfolgreich absolviert, wenn mindestens 4 der darin aufgelistete Module erfolgreich absolviert wurden.
- Die absolvierten Studienschwerpunkte werden auf einem separaten Anhang des Abschlusszeugnisses dargestellt, bei mehr als einem auf Antrag an das Prüfungsamt auch nur in Teilen.
- Auf Antrag kann ein Studienschwerpunkt um weitere passende Module ergänzt werden. Ein solcher Antrag ist bis spätestens sechs Monate vor einer geplanten Teilnahme an einem zu ergänzenden Modul formlos an die Studiengangsleitung zu richten. Über die Annahme des Antrags befindet der Prüfungsausschuss im Benehmen mit der Studiengangsleitung und fachlich geeigneten Lehrpersonen.

8.1 ES - Eingebettete Systeme

Der Schwerpunkt "Eingebettete Systeme" befasst sich mit Informationstechnik, die in Objekte aller Art integriert wird und dort in Analogie zu biologischen Systemen das Nervensystem des Geräts bildet. Viele Gegenstände des Alltags sind inzwischen mit eingebetteten Systemen ausgerüstet, von der Kaffeemaschine über alle Arten von Fahrzeugen bis hin zu Gebäuden, nicht zuletzt auf Grund der hohen Verfügbarkeit und niedrigen Kosten der Mikrocontrollertechnik. Der Studienschwerpunkt vermittelt Wissen zu eingebetteten Systemen sowie zu Komponenten eingebetteter Systeme (Sensoren, Aktoren) und zu autonomen Systemen.

Module der Fakultät:

Modulkürzel	Modulbezeichnung	ECTS
EL	Elektronik	5
ES	Eingebettete Systeme	5
ESP	Eingebettete Systeme - Projekt	5
SOP	Systems on Programmable Chips	5
ATS	Autonome Systeme	5

8.2 IOT - Internet of Things

Das Internet der Dinge (englisch Internet of Things, Kurzform: IoT) ist ein Sammelbegriff für Technologien einer globalen Infrastruktur der Informationsgesellschaften, die es ermöglicht, physische und virtuelle Gegenstände miteinander zu vernetzen und sie durch Informations- und Kommunikationstechniken zusammenarbeiten zu lassen. (Quelle Wikipedia)

Module der Fakultät:

Modulkürzel	Modulbezeichnung	ECTS
IOT	IoT Protokolle und Anwendungen	5
FIT	Funksysteme für IoT	5
DM	Data Mining	5
SOP	Systems on Programmable Chips	5
QKC	Quellen- und Kanalcodierung	5
SMP	Signalverarbeitung mit Matlab/Python und μ C	5
SMO	Smart Mobility Components	5

8.3 NVS - Netze und Verteilte Systeme

Heute sind Computer meistens vernetzt, also über Kommunikationsnetze, WLANs oder Mobilfunknetze miteinander verbunden. Die Software, die diese Computer ausführen, ist typischerweise verteilt. Der Begriff des vernetzten Computers geht dabei weit über traditionelle Desktop- und Netbook-Computer oder Serversysteme hinaus: Er umfasst beispielsweise auch Mobiltelefone ("Smartphones"), eingebettete Systeme, z.B. in vernetzten Fahrzeugen, oder sensorgestützte Systeme zur Überwachung und Steuerung von Gebäuden und Anlagen ("Smart Home", "Smart Factory"). Die Themengebiete dieses Schwerpunkts erstrecken sich von den Grundlagen der Datenübertragung, über Architekturen und Modelle für Computernetze sowie ihre Realisierung durch Protokollhierarchien, bis hin zur Programmierung von Mobilgeräten und Webanwendungen.

Module der Fakultät:

Modulkürzel	Modulbezeichnung	ECTS
NSA	Netzsicherheit und Automation	5
EKS	Entwicklung komplexer Software-Systeme	5
VMA	Programmierung verteilter und mobiler Anwendungen	5
IOT	IoT Protokolle und Anwendungen	5
SM	Software Management	5

8.4 SMS - Smart Systems

Smart Systems sind „intelligente Systeme“, die in der Lage sind, auf Basis einer Datenanalyse selbstständig Entscheidungen, Prognosen oder Empfehlungen zu erzeugen, die dann sowohl zur autonomen Reaktion eines Systems oder auch als Hilfestellung für den Menschen herangezogen werden können. Smart-Systems kommen in verschiedensten Bereichen zum Einsatz. Im Bereich Automotiv z.B. für Einparkhilfen, Abstandsregeltempomaten oder Spurwechselassistenten. Im Bereich der Medizintechnik z.B. für die Auswertung von Röntgenaufnahmen zur Unterstützung des Arztes. Im industriellen Bereich z.B. zur Überwachung von Produktionsprozesse und zur Qualitätskontrolle. Im kaufmännischen Bereich z.B. zur Analyse von Kundenbestellungen zur automatischen Generierung kundenspezifischer Werbemaßnahmen.

Module der Fakultät:

Modulkürzel	Modulbezeichnung	ECTS
ATS	Autonome Systeme	5
ML	Maschinelles Lernen	5
DM	Data Mining	5
CG	Computergrafik	5
CA	Computeranimation	5
VDS	Verteilte Datenverarbeitungssysteme	5

8.5 SOS - Software-Systeme

In der heutigen Informationsgesellschaft wird in fast allen Bereichen komplexe Anwendungssoftware eingesetzt. Die Anwendungen treten dabei in unterschiedlichen Rollen auf: als zentrale Applikationen auf einem Server, als leistungsstarke Klienten, als autonome Systeme sowie als verteilte Anwendungen. In diesem hochdynamischen Umfeld entwickeln sich ständig neue Trends und interessante Technologien. Um solche komplexen Software-Systeme zu realisieren und zu betreiben, sind vielfältige Kenntnisse, Fähigkeiten und Fertigkeiten erforderlich. Die im Schwerpunkt Software-Systeme behandelten Themengebiete umfassen alle Bereiche moderner Anwendungssysteme: Architektur und Design komplexer Software-Systeme, intuitive Benutzerführung, effektive Realisierung der Geschäftslogik, effiziente Datenspeicherung, Programmierung mobiler und autonomer Geräte.

Module der Fakultät:

Modulkürzel	Modulbezeichnung	ECTS
DB2	Datenbanken 2	5
EKS	Entwicklung komplexer Software-Systeme	5
SM	Software Management	5
ATS	Autonome Systeme	5
VMA	Programmierung verteilter und mobiler Anwendungen	5

9. Prüfungsformen

Im Folgenden werden die in den Modulbeschreibungen referenzierten Prüfungsformen näher erläutert. Die Erläuterungen stammen aus der Prüfungsordnung, §19ff. Bei Abweichungen gilt der Text der Prüfungsordnung.

(elektronische) Klausur

Schriftliche, in Papierform oder digital unterstützt abgelegte Prüfung. Genauerer regelt §19 der Prüfungsordnung.

Mündliche Prüfung

Mündlich abzulegende Prüfung. Genauerer regelt §21 der Prüfungsordnung.

Mündlicher Beitrag

Siehe §22, Abs. 5 der Prüfungsordnung: Ein mündlicher Beitrag (z. B. Referat, Präsentation, Verhandlung, Moderation) dient der Feststellung, ob die Studierenden befähigt sind, innerhalb einer vorgegebenen Frist eine praxisorientierte Aufgabe nach wissenschaftlichen und fachpraktischen Methoden selbstständig zu bearbeiten und mittels verbaler Kommunikation fachlich angemessen darzustellen. Dies beinhaltet auch, Fragen des Auditoriums zur mündlichen Darstellung zu beantworten. Die Dauer des mündlichen Beitrags wird von der Prüferin beziehungsweise dem Prüfer zu Beginn des Semesters festgelegt. Die für die Benotung des mündlichen Beitrags maßgeblichen Tatsachen sind in einem Protokoll festzuhalten, zur Dokumentation sollen die Studierenden ebenfalls die schriftlichen Unterlagen zum mündlichen Beitrag einreichen. Die Note ist den Studierenden spätestens eine Woche nach dem mündlichen Beitrag bekanntzugeben.

Fachgespräch

Siehe §22, Abs. 8 der Prüfungsordnung: Ein Fachgespräch dient der Feststellung der Fachkompetenz, des Verständnisses komplexer fachlicher Zusammenhänge und der Fähigkeit zur analytischen Problemlösung. Im Fachgespräch haben die Studierenden und die Prüfenden in etwa gleiche Redeanteile, um einen diskursiven fachlichen Austausch zu ermöglichen. Semesterbegleitend oder summarisch werden ein oder mehrere Gespräche mit einer Prüferin oder einem Prüfer geführt. Dabei sollen die Studierenden praxisbezogene technische Aufgaben, Problemstellungen oder Projektvorhaben aus dem Studiengang vorstellen und erläutern sowie die relevanten fachlichen Hintergründe, theoretischen Konzepte und methodischen Ansätze zur Bearbeitung der Aufgaben darlegen. Mögliche Lösungsansätze, Vorgehensweisen und Überlegungen zur Problemlösung sind zu diskutieren und zu begründen. Die für die Benotung des Fachgesprächs maßgeblichen Tatsachen sind in einem Protokoll festzuhalten.

Projektarbeit

Siehe §22, Abs. 6 der Prüfungsordnung: Die Projektarbeit ist eine Prüfungsleistung, die in der selbstständigen Bearbeitung einer spezifischen Fragestellung unter Anleitung mit wissenschaftlicher Methodik und einer Dokumentation der Ergebnisse besteht. Bewertungsrelevant sind neben der Qualität der Antwort auf die Fragestellung auch die organisatorische und kommunikative Qualität der Durchführung, wie z.B. Slides, Präsentationen, Meilensteine, Projektpläne, Meetingprotokolle usw.

Praktikumsbericht

Siehe §22, Abs. 10 der Prüfungsordnung: Ein Praktikumsbericht (z. B. Versuchsprotokoll) dient der Feststellung, ob die Studierenden befähigt sind, innerhalb einer vorgegebenen Frist eine laborpraktische Aufgabe selbstständig sowohl praktisch zu bearbeiten als auch Bearbeitungsprozess und Ergebnis schriftlich zu dokumentieren, zu bewerten und zu reflektieren. Vor der eigentlichen Versuchsdurchführung können vorbereitende Hausarbeiten erforderlich sein. Während oder nach der Versuchsdurchführung können Fachgespräche stattfinden. Praktikumsberichte können auch in Form einer Gruppenarbeit zur Prüfung zugelassen werden. Die Bewertung des Praktikumsberichts ist den Studierenden spätestens sechs Wochen nach Abgabe des Berichts bekanntzugeben.

Übungspraktikum

Siehe §22, Abs. 11 der Prüfungsordnung: Mit der Prüfungsform "Übungspraktikum" wird die fachliche Kompetenzen bei der Anwendung der in der Vorlesung erlernten Theorien und Konzepte sowie praktische Fertigkeiten geprüft, beispielsweise der Umgang mit Entwicklungswerkzeugen und Technologien. Dazu werden semesterbegleitend mehrere Aufgaben gestellt, die entweder alleine oder in Gruppenarbeit, vor Ort oder auch als Hausarbeit bis zu einem jeweils vorgegebenen Termin zu lösen sind. Die Lösungen der Aufgaben sind durch die Studierenden in (digitaler) schriftlicher Form einzureichen. Die genauen Kriterien zum Bestehen der Prüfung wird zu Beginn der entsprechenden Lehrveranstaltung bekannt gegeben.

Übungspraktikum unter Klausurbedingungen

Siehe §22, Abs. 11, Satz 5 der Prüfungsordnung: Ein "Übungspraktikum unter Klausurbedingungen" ist ein Übungspraktikum, bei dem die Aufgaben im zeitlichen Rahmen und den Eigenständigkeitsbedingungen einer Klausur zu bearbeiten sind.

Hausarbeit

Siehe §22, Abs. 3 der Prüfungsordnung: Eine Hausarbeit (z.B. Fallstudie, Recherche) dient der Feststellung, ob die Studierenden befähigt sind, innerhalb einer vorgegebenen Frist eine Fachaufgabe nach wissenschaftlichen und fachpraktischen Methoden selbstständig in schriftlicher oder elektronischer Form zu bearbeiten. Das Thema und der Umfang (z. B. Seitenzahl des Textteils) der Hausarbeit werden von der Prüferin beziehungsweise dem Prüfer zu Beginn des Semesters festgelegt. Eine Eigenständigkeitserklärung muss vom Prüfling unterzeichnet und abgegeben werden. Zusätzlich können Fachgespräche geführt werden.

Lernportfolio

Ein Lernportfolio dokumentiert den studentischen Kompetenzentwicklungsprozess anhand von Präsentationen, Essays, Ausschnitten aus Praktikumsberichten, Inhaltsverzeichnissen von Hausarbeiten, Mitschriften, To-Do-Listen, Forschungsberichten und anderen Leistungsdarstellungen und Lernproduktionen, zusammengefasst als sogenannte „Artefakte“. Nur in Verbindung mit der studentischen Reflexion (schriftlich, mündlich oder auch in einem Video) der Verwendung dieser Artefakte für das Erreichen des zuvor durch die Prüferin oder den Prüfer transparent gemachten Lernziels wird das Lernportfolio zum Prüfungsgegenstand. Während der Erstellung des Lernportfolios wird im Semesterverlauf Feedback auf Entwicklungsschritte und/oder Artefakte gegeben. Als Prüfungsleistung wird eine nach dem Feedback überarbeitete Form des Lernportfolios - in handschriftlicher oder elektronischer Form - eingereicht.

Schriftliche Prüfung im Antwortwahlverfahren

Siehe §20 der Prüfungsordnung.

Zugangskolloquium

Siehe §22, Abs. 12 der Prüfungsordnung: Ein Zugangskolloquium dient der Feststellung, ob die Studierenden die versuchsspezifischen Voraussetzungen erfüllen, eine definierte laborpraktische Aufgabe nach wissenschaftlichen und fachpraktischen Methoden selbstständig und sicher bearbeiten zu können.

Testat / Zwischentestat

Siehe §22, Abs. 7 der Prüfungsordnung: Mit einem Testat/Zwischentestat wird bescheinigt, dass die oder der Studierende eine Studienarbeit (z.B. Entwurf) im geforderten Umfang erstellt hat. Der zu erbringende Leistungsumfang sowie die geforderten Inhalte und Anforderungen ergeben sich aus der jeweiligen Modulbeschreibung im Modulhandbuch sowie aus der Aufgabenstellung.

Open-Book-Ausarbeitung

Die Open-Book-Ausarbeitung oder -Arbeit (OBA) ist eine Kurz-Hausarbeit und damit eine unbeaufsichtigte schriftliche oder elektronische Prüfung. Sie zeichnet sich dadurch aus, dass gemäß Hilfsmittelerklärung der Prüferin bzw. des Prüfers in der Regel alle Hilfsmittel zugelassen sind. Auf die Sicherung guter wissenschaftlicher Praxis durch ordnungsgemäßes Zitieren etc. und das Erfordernis der Eigenständigkeit der Erbringung jedweder Prüfungsleistung wird besonders hingewiesen.

Abschlussarbeit

Bachelor- oder Masterarbeit im Sinne der Prüfungsordnung §25ff.: Die Masterarbeit ist eine schriftliche Hausarbeit. Sie soll zeigen, dass die oder der Studierende befähigt ist, innerhalb einer vorgegebenen Frist ein Thema aus ihrem oder seinem Fachgebiet sowohl in seinen fachlichen Einzelheiten als auch in den fachübergreifenden Zusammenhängen nach wissenschaftlichen und fachpraktischen Methoden selbstständig zu bearbeiten. Die interdisziplinäre Zusammenarbeit kann auch bei der Abschlussarbeit berücksichtigt werden.

Kolloquium

Kolloquium zur Bachelor- oder Masterarbeit im Sinne der Prüfungsordnung §29: Das Kolloquium dient der Feststellung, ob die Studentin oder der Student befähigt ist, die Ergebnisse der Masterarbeit, ihre fachlichen und methodischen Grundlagen, fachübergreifende Zusammenhänge und außerfachlichen Bezüge mündlich darzustellen, selbstständig zu begründen und ihre Bedeutung für die Praxis einzuschätzen.

10. Profil-Modulmatrix

Im Folgenden wird dargestellt, inwieweit die Module des Studiengangs die Kompetenzen und Handlungsfelder des Studiengangs sowie hochschulweite Studiengangskriterien stützen bzw. ausbilden.

Kürzel	Modulbezeichnung	HF1 - Systeme zur Verarbeitung,...	HF2 - Anforderungen, Konzepte u...	HF3 - Informationstechnische Sy...	HF4 - Mit Auftraggebern, Anwend...	K.1 - In Systemen denken	K.2 - fachliche Probleme abstra...	K.3 - Konzepte und Methoden der...	K.4 - Systeme analysieren	K.5 - Systeme entwerfen	K.6 - Systeme realisieren	K.7 - Systeme prüfen	K.8 - Informationen beschaffen ...	K.9 - Typische Werkzeuge, Stand...	K.10 - In vorhandene Systeme ein...	K.11 - Projekte organisieren	K.12 - Grundzüge wissenschaftlic...	K.13 - Komplexe technische Aufga...	K.14 - Gesellschaftliche und eth...	K.15 - Befähigung zum lebenslang...	K.16 - Kommunikative und interku...	SK.1 - Global Citizenship	SK.2 - Internationalisierung	SK.3 - Interdisziplinarität	SK.4 - Transfer
AD	Algorithmen und Datenstrukturen	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●		●	●	●				
ASN	Angewandte Statistik und Numerik	●	●	●	●	●	●	●	●	●			●							●					
AT	Antennentechnik	●	●	●	●				●		●	●	●	●		●	●			●	●				
ATS	Autonome Systeme	●		●	●	●	●	●	●	●	●	●	●							●					
BAA	Bachelorarbeit	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●			
BVS1	Betriebssysteme	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●		●	●	●				
BVS2	Verteilte Systeme	●	●	●	●	●	●	●	●	●	●	●	●			●	●			●					
BWR	Betriebswirtschaft und Recht				●		●										●		●	●		●		●	
CA	Computeranimation	●		●	●	●	●	●	●	●	●	●	●							●				●	●
CG	Computergrafik	●			●	●	●	●	●		●	●	●						●	●				●	●
DB1	Datenbanken 1	●	●	●	●	●	●	●	●	●	●			●							●				
DB2	Datenbanken 2	●	●	●	●	●	●				●	●													
DM	Data Mining	●		●	●		●		●				●					●	●	●					
DR	Digitalrechner	●		●	●	●	●	●	●	●	●	●	●		●	●			●	●					
EG	Elektrotechnische Grundlagen für die Technische Informatik	●	●			●	●	●	●				●					●		●	●				
EKS	Entwicklung komplexer Software-Systeme	●	●	●	●	●	●	●	●	●	●	●	●	●					●	●	●				
EL	Elektronik	●	●	●	●	●	●		●	●	●	●		●		●			●	●					
ES	Eingebettete Systeme	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●				

Kürzel	Modulbezeichnung	HF1 - Systeme zur Verarbeitung,...	HF2 - Anforderungen, Konzepte u...	HF3 - Informationstechnische Sy...	HF4 - Mit Auftraggebern, Anwend...	K.1 - In Systemen denken	K.2 - fachliche Probleme abstra...	K.3 - Konzepte und Methoden der...	K.4 - Systeme analysieren	K.5 - Systeme entwerfen	K.6 - Systeme realisieren	K.7 - Systeme prüfen	K.8 - Informationen beschaffen ...	K.9 - Typische Werkzeuge, Stand...	K.10 - In vorhandene Systeme ein...	K.11 - Projekte organisieren	K.12 - Grundzüge wissenschaftlic...	K.13 - Komplexe technische Aufga...	K.14 - Gesellschaftliche und eth...	K.15 - Befähigung zum lebenslang...	K.16 - Kommunikative und interku...	SK.1 - Global Citizenship	SK.2 - Internationalisierung	SK.3 - Interdisziplinarität	SK.4 - Transfer
ESP	Eingebettete Systeme - Projekt	●	●	●			●		●	●	●	●			●	●	●		●	●					
FIT	Funksysteme für IoT	●	●	●	●		●	●	●							●	●								
FSA	Formale Sprachen und Automatentheorie	●	●	●	●	●	●	●	●	●	●		●												
GRT	Graphentheorie	●					●		●																
GSP	Grundlagen der Systemprogrammierung	●	●	●	●	●	●	●	●	●	●	●	●	●					●	●	●				
GUI	Graphische Oberflächen und Interaktion	●	●	●		●	●			●	●	●				●	●			●					
HF	Hochfrequenztechnik	●	●	●	●				●	●				●											
IAK	Ingenieurakustik	●	●	●	●	●		●		●									●						
IBV	Industrielle Bildverarbeitung	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●						
IOT	IoT Protokolle und Anwendungen	●	●	●	●	●	●	●	●	●	●														
IPM	IT-Projektmanagement	●	●	●	●	●			●					●		●									
ITS	IT-Sicherheit	●	●	●	●					●	●	●						●							
KOAK	Kommunikationsakustik	●	●	●	●	●		●	●	●									●						
KOLL	Kolloquium zur Bachelorarbeit													●						●		●		●	●
MA1	Mathematik 1	●	●	●		●			●					●											
MA2	Mathematik 2	●	●	●		●			●					●											
ML	Maschinelles Lernen	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●						
MLO	Machine Learning Operations	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●						
MT	Messtechnik	●	●	●	●	●	●	●	●		●					●	●	●							

Kürzel	Modulbezeichnung	HF1 - Systeme zur Verarbeitung,...	HF2 - Anforderungen, Konzepte u...	HF3 - Informationstechnische Sy...	HF4 - Mit Auftraggebern, Anwend...	K.1 - In Systemen denken	K.2 - fachliche Probleme abstra...	K.3 - Konzepte und Methoden der...	K.4 - Systeme analysieren	K.5 - Systeme entwerfen	K.6 - Systeme realisieren	K.7 - Systeme prüfen	K.8 - Informationen beschaffen ...	K.9 - Typische Werkzeuge, Stand...	K.10 - In vorhandene Systeme ein...	K.11 - Projekte organisieren	K.12 - Grundzüge wissenschaftlic...	K.13 - Komplexe technische Aufga...	K.14 - Gesellschaftliche und eth...	K.15 - Befähigung zum lebenslang...	K.16 - Kommunikative und interku...	SK.1 - Global Citizenship	SK.2 - Internationalisierung	SK.3 - Interdisziplinarität	SK.4 - Transfer
NDQ	Nachhaltigkeit durch Qualität		●		●		●	●				●		●		●	●	●			●			●	●
NP	Netze und Protokolle	●	●	●	●	●	●	●	●	●	●	●	●	●					●	●	●				
NSA	Netzsicherheit und Automation	●	●	●	●	●	●	●	●	●	●	●	●	●					●	●	●				
PHTB	Philosophische Handlungstheorie Bachelor																					●		●	
PI1	Praktische Informatik 1	●		●	●		●		●		●	●	●	●		●	●		●	●	●				
PI2	Praktische Informatik 2	●		●	●		●	●	●		●	●													
PP	Programmierpraktikum	●		●	●	●	●	●	●		●	●	●	●		●	●		●	●	●				
PPRA	Parallelprogrammierung und Rechnerarchitekturen	●		●	●	●	●	●	●	●	●	●	●	●					●	●	●				
PRA	Praxisphase	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
PUK	Präsentation und Kommunikation												●	●				●		●	●	●		●	
QKC	Quellen- und Kanalcodierung	●		●	●	●	●	●	●	●															
RT	Regelungstechnik	●	●	●	●	●	●	●	●	●	●	●		●											
SE	Software Engineering	●	●	●	●	●	●	●	●	●	●	●													
SIG	Signalverarbeitung	●	●	●	●	●	●	●	●	●	●														
SM	Software Management	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●		●					
SMO	Smart Mobility Components	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●				●	●
SMP	Signalverarbeitung mit Matlab/Python und µC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●				
SOP	Systems on Programmable Chips	●		●	●	●		●	●	●	●	●				●	●								

Kürzel	Modulbezeichnung	HF1 - Systeme zur Verarbeitung,...	HF2 - Anforderungen, Konzepte u...	HF3 - Informationstechnische Sy...	HF4 - Mit Auftraggebern, Anwend...	K.1 - In Systemen denken	K.2 - fachliche Probleme abstra...	K.3 - Konzepte und Methoden der...	K.4 - Systeme analysieren	K.5 - Systeme entwerfen	K.6 - Systeme realisieren	K.7 - Systeme prüfen	K.8 - Informationen beschaffen ...	K.9 - Typische Werkzeuge, Stand...	K.10 - In vorhandene Systeme ein...	K.11 - Projekte organisieren	K.12 - Grundzüge wissenschaftlic...	K.13 - Komplexe technische Aufga...	K.14 - Gesellschaftliche und eth...	K.15 - Befähigung zum lebenslang...	K.16 - Kommunikative und interku...	SK.1 - Global Citizenship	SK.2 - Internationalisierung	SK.3 - Interdisziplinarität	SK.4 - Transfer
SWP	Softwarepraktikum	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●		●	●	●				
SYP	Systementwicklungs-Projekt	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				
UT	Übertragungstechnik	●	●	●	●	●	●	●	●	●			●						●	●					
VDS	Verteilte Datenverarbeitungssysteme	●	●	●	●	●	●	●	●	●	●	●	●			●	●			●					
VMA	Programmierung verteilter und mobiler Anwendungen	●	●		●		●				●	●	●	●	●			●	●	●	●				
XIB	Fachübergreifende Kompetenzen und Soft-Skills																								
XIB2	Fachübergreifende Kompetenzen und Soft-Skills																								
XPSS	Praxisorientierte Summer School	●	●		●	●									●						●				

11. Versionsverlauf

In untenstehender Tabelle sind die verschiedenen Versionen des Lehrangebots aufgeführt. Die Versionen sind umgekehrt chronologisch sortiert mit der aktuell gültigen Version in der ersten Zeile. Die einzelnen Versionen können über den Link in der rechten Spalte aufgerufen werden.

Version	Datum	Änderungen	Link
3.10	2024-11-29-12-00-00	1. Version zur Abstimmung im Fakultätsrat	Link
3.9	2024-11-29-12-00-00	1. Begutachtete Version für Reakkreditierung 2024 2. Neues Layout für sämtliche Modulhandbücher	Link
3.8	2024-06-11-14-00-00	1. Mitarbeit in Gremien im Bachelor/Master Technische Informatik und Bachelor Elektrotechnik 2. Produktentwicklung für Smart-Cities im Wintersemester (verantwortlich jetzt Prof. Humpert) 3. Überarbeitung Modul und Lehrveranstaltung Elektrische Sicherheit und EMV 4. Überarbeitung Modul und Lehrveranstaltung Hochspannungstechnik 5. Überarbeitung Modul und Lehrveranstaltung Hochspannungsübertragungstechnik 6. Neue(s) Modul und Lehrveranstaltung Themen der Medientechnologie	Link
3.7	2024-05-10-14-30-00	1. Weitere Literaturangabe in Lehrveranstaltung Antennentechnik 2. Medienrecht: Tutorium entfernt, Prüfungstyp zu "andere studienbegleitende Prüfungsform" geändert, weitere Literaturangaben, Modul in Bachelor Medientechnologie angelegt 3. Änderungen der Dozenten und Literatur in Lehrveranstaltung "Informatik 1/2", "Mensch-Maschine-Interaktion" und "Parallelprogrammierung" 4. Lehrveranstaltung Leistungselektronik: Überarbeitung der Prüfungsdetails und Literaturangaben 5. Änderung der Modulverantwortung von Kunz zu Reiter / Grünvogel / Salmen in "Visuelle und auditive Wahrnehmung", "Mathematik 2" und "Signaltheorie und angewandte Mathematik" 6. Anlage Modul "Interdisziplinäres Projekt" im Bachelor Medientechnologie 7. Lehrveranstaltung Energiemanagement in Energieverbundsystemen: Änderungen an Präsenzzeit, Räumen und Prüfungstyp (jetzt Projektarbeit + E-Assessment) 8. Lehrveranstaltung Energiespeicher: Streichung des Tutoriums, Prüfungstyp jetzt Fachgespräch 9. Lehrveranstaltung Energiewirtschaft: Überarbeitung der Fertigungsbeschreibungen und des Begleitmaterials, Streichung des Tutoriums 10. Modul SmartCity: Änderung Modulverantwortung von Stadler zu Humpert 11. Lehrveranstaltung Elektrische Antriebe: Überarbeitung der Prüfungsbeschreibung und Praktikumskompetenzen 12. Lehrveranstaltung Schaltzetteile: Überarbeitungen der Prüfungsbeschreibung, -typ, Begleitmaterialien, Präsenzlehre und Fertigkeiten. 13. Leistungselektronische Stellglieder für PV- und Windkraftanlagen: Überarbeitungen der Prüfungsbeschreibung, -typ, Begleitmaterialien, Präsenzlehre und Fertigkeiten.	Link
3.6	2024-02-23-15-00-00	1. Generelle Überarbeitung des Layouts 2. Eingangstexte bei Wahlmodulkatalogen und Schwerpunkten überarbeitet und POs angeglichen 3. Lehrveranstaltung BWR (Kim) sowohl im Sommer- als auch Wintersemester. 4. Modellierung von Energiesystemen der Fakultät 09 als wählbares Modul im allgemeinen Wahlkatalog im Master Technische Informatik	Link
3.5	2023-07-20-15-00-00	1. Neue Lehrveranstaltung "Projektleitung Systementwurfs-Praktikum" im Master Technische Informatik (anerkannt für XIM) 2. Geänderter Studienverlauf mit vermindertem Workload in Bachelor Technische Informatik	Link

Version	Datum	Änderungen	Link
3.4	2023-04-17-15-00-00	1. Modul "Netze und Automation (NSA)" (vormals "Internetworking (IN)") den Schwerpunkten "Internet of Things" und "Information- und Kommunikationstechnik" des Studiengangs "Bachelor Elektrotechnik" hinzugefügt (FR-2023-6) 2. Modulkürzel für "Autonome Systeme" lautet jetzt auch in "Bachelor Technische Informatik" ATS (FR-2023-7)	Link
3.3	2023-03-27-11-00-00	1. Lehrveranstaltung Grundgebiete der Elektrotechnik 2 (Kronberger) aktualisiert; Modul "Entrepreneurship, Gewerblicher Rechtsschutz, Market Knowledge" auch für X1 in Master Technische Informatik wählbar.	Link
3.2	2023-03-08-16-00-00	1. Licht- und Beleuchtungstechnik (LB) Wintersemester-Wahlmodulen zugeordnet; BaTIN-Modul "Web-Architekturen" SGL zugeordnet und vakante Lehrveranstaltung vorerst entfernt.	Link
3.1	2023-02-24-20-00-00	1. Allgemeine Bereinigung von kaputten Links (http 404)	Link
3.0	2022-12-09-12-31-39	1. Umbenennung des Moduls und der Lehrveranstaltung "Internetworking und Netzsicherheit (IN)" in "Netzsicherheit und Automation (NSA) (FR-2022-20)"	Link

Impressum

Datenschutzhinweis

Haftungshinweis

Bei Fehlern, bitte Mitteilung an
die
modulhandbuchredaktion@f07.th-koeln.de