



Modulhandbuch

Bachelor Studiengang Automobilinformatik (B.Sc.)

Hochschule Landshut
gültig ab dem Sommersemester 2026

beschlossen am 28. Juli 2026

Hinweis:

Die auf Basis der SPO konkret zu erbringende Prüfungsleistung wird bis spätestens zwei Wochen nach Beginn des Semesters durch die Fakultät Informatik im Studien- und Prüfungsplan hochschulöffentlich bekannt gegeben.

Inhaltsverzeichnis

Beispielhafter Studienverlauf	3
AIF110 Grundlagen der Informatik	4
AIF111 Programmieren I	5
AIF112 Digitaltechnik	7
AIF211 Programmieren II	8
AIF212 Software Engineering	9
AIF311 Datenbanken	11
AIF312 Modellbasierte Entwicklung I	12
AIF370 IT-Sicherheit	14
AIF390 Praxisorientiertes Studienprojekt	15
AIF410 Echtzeitbetriebssysteme	16
AIF411 Algorithmen und Datenstrukturen	18
AIF412 Datenkommunikation	19
AIF413 Modellbasierte Entwicklung II	21
AIF590 Praktische Zeit im Betrieb	22
AIF591 Praxisseminar	23
AIF612 Softwarearchitekturen	24
AIF620 Entwicklung sicherheitskritischer Systeme	26
AIF630 Autonome Fahrzeuge	28
AIF670 Fachbezogenes Wahlpflichtmodul	30
AIF710 Prozessrechentechnik	31
AIF720 Seminar	33
AIF790 Bachelor-Arbeit	34
Module aus dem Bachelor-Studiengang Automobilwirtschaft und -technik	35

Grundlagen der Informatik

AIF110

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Prof. Dr. Dieter Nazareth
Studiengang:	Bachelor
Modultyp:	Pflichtmodul
Sprache:	Deutsch
Angebot:	im ersten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	-
Voraussetzungen:	-
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit im Unterricht 90 Stunden Selbststudium
Lehrformen:	4 SWS seminaristischer Unterricht
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben. Die Prüfung zu diesem Modul ist Bestandteil der Grundlagen- und Orientierungsprüfung und muss spätestens am Ende des zweiten Studiensemesters angetreten werden.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden haben einen Überblick über wichtige Gebiete der Informatik und vertiefte Kenntnisse in ausgewählten Bereichen. Sie kennen die Prinzipien verschiedenartiger Programmiersprachen und Datenstrukturen und sind in der Lage, darauf basierend einfache Algorithmen zu erstellen. Ferner können Sie diese Algorithmen hinsichtlich ihrer Effizienz bewerten.

Lehrinhalte:

- Informationssysteme
- Kodierung
- Informelle Algorithmen
- Textersetzung
- Struktogramme
- Funktionale Programmiersprachen
- Prozedurale Programmiersprachen
- Statische Datentypen
- Dynamische Datentypen
- Referenzen
- Komplexität und Berechenbarkeit

Literatur:

M. Broy: Informatik 1: Programmierung und Rechnerstrukturen. Springer-Verlag, Berlin 1997.
H.-P. Grumm, M. Sommer: Einführung in die Informatik, Oldenbourg Verlag, 2004.

Programmieren I

AIF111

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Prof. Dr. Dieter Nazareth
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im ersten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	-
Voraussetzungen:	-
Leistungspunkte:	7
Arbeitsaufwand:	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit Praktikum 120 Stunden Selbststudium
Lehrformen:	4 SWS seminaristischer Unterricht 2 SWS Praktikum
Leistungsnachweise und Prüfung:	Leistungsnachweis im Praktikum, Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben. Die Prüfung zu diesem Modul ist Bestandteil der Grundlagen- und Orientierungsprüfung und muss spätestens am Ende des zweiten Studiensemesters angetreten werden.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden kennen angemessene Verfahren, Methoden und Algorithmen zur Lösung von Problemstellungen einfachen bis mittleren Umfangs in den Programmiersprachen C und C++. Sie sind in der Lage theoretisch erworbenes Wissen planmäßig und systematisch in lauffähige, effiziente Software umzusetzen, die Lösungen angemessen zu testen, sowie strukturelle Schwachstellen zu erkennen und zu beseitigen. Die Studierenden haben verstanden, dass die entwickelten Lösungen -modular, flexibel und kompakt strukturiert sein müssen.

Lehrinhalte:

- Elementare Datentypen
- Grundlegende Konzepte der prozeduralen Programmierung
- Grundlegende Konzepte der funktionalen Programmierung
- Präprozessoranweisungen
- Ein- und Ausgabe in C und C++
- Datenstrukturen
- Zeiger, Vektoren und Felder
- Grundlegende Konzepte der objektorientierten Programmierung
- Vererbung, Mehrfachvererbung

Literatur:

Brian W. Kernighan, Dennis Ritchie: The C Programming Language, Prentice Hall Software, aktuellste Auflage

Jürgen Wolf: C von A bis Z: Das umfassende Handbuch, Galileo Computing, aktuellste Ausgabe

Jürgen Wolf: C++: Das umfassende Handbuch, aktuell zum Standard C++11, Galileo Computing, aktuellste Auflage

Digitaltechnik

AIF112

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Prof. Dr. Jürgen Welter
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im ersten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	-
Voraussetzungen:	-
Leistungspunkte:	3
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 60 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht und Übungen
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden kennen wichtige Schaltnetze und Schaltwerke, die als Grundbausteine in Mikroprozessoren verwendet werden. Sie haben die Fähigkeit einfache Schaltwerke zu entwerfen.

Lehrinhalte:

- Schaltalgebra und Schaltsymbole
- Schaltnetze (Kombinatorische Logik):
- Codeumsetzer, Decoder, Demultiplexer, Multiplexer
- Arithmetische Schaltnetze:
- Addierer, Subtrahierer, Arithmetisch-logische Einheit (ALU), Multiplizierer
- Flip-Flops:
- Ungetaktete (asynchrone) Flip-Flops, Getaktete (synchrone) Flip-Flops
- Schaltwerke (Sequentielle Logik):
- Zähler, Register, Schieberegister
- Busse
- Halbleiterspeicher
- Grundstruktur eines Mikroprozessors

Literatur:

S. Tanenbaum: „Structured Computer Organization“, Prentice Hall, 2005.
 Borgmeyer: Grundlagen der Digitaltechnik, Hanser, 2001.
 Fricke, Digitaltechnik, Vieweg, 2005

Programmieren II

AIF211

Modulverantwortlicher:	Prof. Dr. Christopher Auer
Dozent:	Prof. Dr. Christopher Auer
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im zweiten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I
Voraussetzungen:	-
Leistungspunkte:	7
Arbeitsaufwand:	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit Praktikum 120 Stunden Selbststudium
Lehrformen:	4 SWS seminaristischer Unterricht 2 SWS Praktikum (jeweils 14-tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Leistungsnachweis im Praktikum, Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden können angemessene Verfahren, Methoden und Algorithmen zur Lösung von Problemstellungen einfachen bis mittleren Umfangs in der Programmiersprache Java anwenden. Sie sind in der Lage theoretisch erworbenes Wissen planmäßig und systematisch in lauffähige, effiziente Software umzusetzen, sowie strukturelle Schwachstellen zu erkennen und zu beseitigen. Die Studierenden erwerben ein Verständnis dafür, wie Softwarelösungen modular, flexibel und kompakt zu gestalten sind.

Lehrinhalte:

- Java Laufzeitsystem, Garbage Collection
- Java Typsystem
- Konzepte der objektorientierten Programmierung, Vererbung und Beziehungen zwischen Typen.
- Datenkapselung, Immutability, Konzepte von Gleichheit und Identität
- Entwicklung von Lösungen für konkrete Problemstellungen und Umsetzung der Lösungsideen in lauffähige Software unter Einhaltung professioneller Maßstäbe und Kriterien
- Einsatz von Klassenbibliotheken und Umgang mit Fehlern
- Definition und Nutzung von Container-Datenstrukturen

Literatur:

Christian Ullentbloom: Java ist auch eine Insel. Rheinwerk Computing 2023.
 Michael Inden: Der Weg zum Java-Profi, dpunkt.Verlag 2020
 Herbert Prähofer: Funktionale Programmierung in Java. dpunkt.verlag GmbH 2020.

Software Engineering

AIF212

Modulverantwortlicher:	Prof. Dr. Sebastian Schröter
Dozent:	Prof. Dr. Sebastian Schröter
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im vierten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	-
Voraussetzungen:	-
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit in den Übungen 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Übungen
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden werden für das Thema Software Engineering motiviert. Sie erlangen ein Verständnis für Softwarequalität und erhalten einen Überblick über alle Lebenszyklusphasen der Softwareentwicklung. Die Studierenden kennen verschiedene Vorgehensmodelle bei der Softwareentwicklung. Die Studierenden erlangen grundlegende Kenntnisse über die Aktivitäten in der Anforderungsanalyse, Architektur, im Entwurf und Test. Die Studierenden kennen die Bedeutung von Soft Skills im Entwicklungsprozess. In den Übungen werden gemeinsam konkrete Fragestellungen beantwortet und ausgesuchte Beispiele bearbeitet. Außerdem wird Bezug zur Bedeutung von nachhaltigem Handeln hergestellt mit Referenz auf die SDGs 7-12.

Lehrinhalte:

Motivation und Definition der Begriffe Software, Software Engineering, Softwarequalität usw., Planung (Projektplanung, Aufwandsschätzung, Machbarkeitsstudie, Lastenheft), Anforderungsanalyse (Stakeholder, Kommunikation, Kontext, Kontextmodellierung, Einsatzszenarios, Qualitätskriterien, Formulierung von Anforderungen, Differenzierung Lasten-/Pflichtenheft), Softwarearchitektur (Grundprinzipien, Begriffe, Muster), Entwurf (Datenmodellierung, Zustandsmodellierung, Entity-Relationship Diagramme, Muster), Testen (Verifikation und Validierung, statisches und dynamisches Testen, Äquivalenzklassentest, Kontrollflussgraphen, Whitebox Testen) Entscheidungsprozess, Konfigurationsmanagement, elementare Grundlagen der Qualitätssicherung.

Literatur:

- Helmut Balzert: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Spektrum Akademischer Verlag, 2009.
- Helmut Balzert: Lehrbuch der Software-Technik: Software-Management. Spektrum Akademischer Verlag, 2008.
- Helmut Balzert: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. Spektrum Akademischer Verlag, 2011.
- Ian Sommerville: Software Engineering, 6. Auflage, Verlag Pearson Studium, 2001
- Wolfgang Zuser et al.: Software Engineering mit UML und dem Unified Process, Verlag Pearson Studium, 2001
- Grady Booch et al.: Das UML-Benutzerhandbuch, Addison-Wesley, 1999
- Grady Booch: Objektorientierte Analyse und Design, 2. Auflage, Addison-Wesley, 1996
- Bernd Oestereich: Objektorientierte Softwareentwicklung, Oldenbourg Verlag, 5. Auflage, 2001 A. Spillner, T. Linz: Basiswissen Softwaretest. dpunkt.verlag, 2012
- Peter Liggesmeyer: Software-Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum Akademischer Verlag, 2009.

Datenbanken

AIF311

Modulverantwortlicher:	M.Sc. Thomas Franzke
Dozent:	M.Sc. Thomas Franzke
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im dritten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Erster Studienabschnitt oder vergleichbare Kenntnisse, Grundkenntnisse in Java.
Voraussetzungen:	
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden besitzen theoretische und praktische Kenntnisse über relationale, objektrelationale und NoSQL-Datenbanken. Sie können Datenbanken entwerfen, aus dem Entwurf ein Schema erstellen und dieses mit Hilfe eines Datenbanksystems umsetzen. Ebenso lernen sie, vorhandene Datenbankentwürfe zu verstehen und bei Bedarf anzupassen, zu erweitern und mit eigener Software auf die Systeme zuzugreifen.

Lehrinhalte:

- Aufbau und Funktionen eines Datenbanksystems
- Datenbankentwurf: Entity-Relationship-Modell, Normalisierung
- Relationales Datenbank-Modell
- Anfragesprachen: relationale Algebra, Structured Query Language (SQL)
- Indexstrukturen in relationalen Datenbanken
- Transaktionen, Trigger, Query-Optimierung
- eingebettetes SQL, Java Database Connectivity (JDBC)
- NoSQL-Datenbanken (MongoDB)

Literatur:

- [1] A. Kemper, André Eickler: Datenbank-Systeme, De Gruyter
- [2] R. Elmasri, S. B. Navathe: Grundlagen von Datenbanksystemen, Addison-Wesley

Modellbasierte Entwicklung I

AIF312

Modulverantwortlicher:	Prof. Dr. Martin Pellkofer
Dozent:	Prof. Dr. Martin Pellkofer
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im dritten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I
Voraussetzungen:	Zulassung zum Praktikum erfolgt bei bestandener Modulprüfung in Programmieren I oder Programmieren II
Leistungspunkte:	7
Arbeitsaufwand:	60 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 120 Stunden Selbststudium
Lehrformen:	4 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden verstehen die Methode der modellbasierten Softwareentwicklung und die Unterschiede zur traditionellen Vorgehensweise. Sie sind in der Lage, mit Hilfe der Werkzeugkette Matlab/Simulink/Stateflow modellbasiert komplexe Fahrzeugfunktionen zu entwickeln und die Funktionen in Festkommaarithmetik zu formulieren. Sie können die Modelle der Fahrzeugfunktionen auf verschiedenen generischen Plattformen ablaufen lassen und über die I/O-Kanäle mit einem äußeren technischen Prozess verbinden. Die Studenten sind in der Lage, aus den Modellen der Fahrzeugfunktionen Quelltext für die Sprache C zu generieren. Sie können dabei den Generierungsprozess so anpassen, dass der Quelltext sich in eine vorgegebene Software-Umgebung auf einem eingebetteten System einfügt.

Lehrinhalte:

- Traditioneller vs. modellbasierter Entwicklungsprozess
- Anforderungen an Modelle und Modellierungstechniken
- Modellierungssprachen und ihre Eigenschaften
- Matlab™: Datentypen, Matrix- und Feldoperationen, Prozeduren und Funktionen, numerisches Lösen von Differentialgleichungen;
- Simulink™: Modellierung dynamischer Systeme durch hierarchische Blockschaltbilder, Stapelverarbeitung von Simulationen mit Variation der Parameter, Erstellen eigener Blockbibliotheken und S-Funktionen, Einbinden von handgeschriebenem Quellcode in das Modell;
- Stateflow™: Ereignisdiskrete Modellierung mit hierarchischen, nebenläufigen Zustandsautomaten
- Automatische Code-Generierung mit Matlab-, Simulink- und Embedded-Coder™
- Modellevolution von Gleitkommaarithmetik zu Festkommaarithmetik mit reversiblen Umschalten zwischen physikalischem Modell und Software-Modell
- Rapid Prototyping mit verschiedenen Plattformen

Literatur:

- A. Angermann, M. Beuschel, M. Rau, W. Wohlfarth: Matlab-Simulink-Stateflow, Oldenbourg, 10. Auflage, München 2021
- P. Marwedel: Eingebettete Systeme, Springer Verlag, Heidelberg, 2008
- F. Tränkle: Modellbasierte Entwicklung Mechatronischer Systeme, De Gruyter, 2021
- K. Berns, B. Schürmann, M. Trapp: Eingebettete Systeme, Vieweg+Teubner, 1. Auflage, Wiesbaden, 2010

IT-Sicherheit

AIF370

Modulverantwortlicher:	Prof. Dr. Johann Uhrmann
Dozent:	Prof. Dr. Johann Uhrmann
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im dritten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I
Voraussetzungen:	-
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit in der Übung 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen (14tägig)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Erkennen von Risiken in der Informationsgesellschaft. Kenntnis wichtiger Dienste und Mechanismen zur Erstellung und zum Einsatz sicherer IT-Systeme.

Lehrinhalte:

Analyse von Sicherheitsbedrohungen.

Die Säulen der IT-Sicherheit: Verfügbarkeit, Vertraulichkeit, Integrität. Sicherheitsbedrohungen.

Sicherheitsbasisdienste: Kryptographie, Key Management, Authentifizierung.

Sicherheitsarchitekturen und Protokolle: pgp, S/MIME, TLS. Firewalls.

Cloud-Security.

Analyse der Sicherheit von Software und Produkten in ihrem Lebenszyklus, Nachhaltigkeitsanalysen.

Aktuelle Entwicklungen in der IT-Sicherheit inkl. deren ethischer Betrachtung und gesellschaftlicher Auswirkung

Literatur:

Roland Hellmann, IT-Sicherheit - Eine Einführung, De Gruyter, 2018.

Michael Messner, Hacking mit Metasploit, dpunkt, 2017.

Claudia Eckert, IT-Sicherheit: Konzepte - Verfahren - Protokolle, De Gruyter, 2023.

Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Praxisorientiertes Studienprojekt

AIF390

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Dozenten der Fakultät Informatik
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	Im sechsten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I, Software Engineering I, Grundlagen der Informatik
Voraussetzungen:	Zulassung erfolgt bei bestandener Prüfung in Programmieren I oder Programmieren II
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden nicht ständig betreute Projektarbeit im Labor 90 Stunden eigenverantwortliches Arbeiten am Projekt
Lehrformen:	4 SWS nicht ständig betreute Projektarbeit im Labor Eigenverantwortliches Arbeiten der Studierenden in Teams von einer kritischen Größe, so dass das Auftreten typischer Schnittstellenprobleme gewährleistet ist.
Leistungsnachweise und Prüfung:	Benoteter Leistungsnachweis durch individuelle schriftliche Ausarbeitung jedes Teammitglieds zum eigenen Beitrag im Projekt, im Team erstellte Gesamtdokumentation.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden kennen die Problematik der Erstellung komplexer Systeme. Sie können die Grundlagen des wissenschaftlichen Arbeitens anwenden und wissen, wie man eigenverantwortlich dem Studiengang entsprechende Projekte durchführt. Sie haben Teamarbeit trainiert und Kenntnisse in der Abschätzung des Umfangs von Projekten sowie in Management und Kontrolle von Projekten erworben. Sie sind in der Lage, fachübergreifende Kenntnisse anzuwenden und Arbeitsergebnisse zu präsentieren.

Lehrinhalte:

Die Lehrenden der Fakultät Informatik bieten den Studierenden per Aushang Projektthemen mit einer kurzen Beschreibung zur Auswahl an. Teams von Studenten können selbst ein Projekt vorschlagen, dafür müssen Sie einen Betreuungsdozenten finden. Die Studenten werden von dem ausgebenden Dozenten regelmäßig fachlich betreut.

Projekte im Rahmen des Moduls "Unternehmerische Kompetenzen" (Campus Company) können ebenfalls anerkannt werden, sofern das Thema des Projekts zum Studiengang passt.

Literatur:

Siehe Projektbeschreibungen.

Echtzeitbetriebssysteme

AIF410

Modulverantwortlicher:	Prof. Dr. Martin Pellkofer
Dozent:	Prof. Dr. Martin Pellkofer
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im vierten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I/II, Modellbasierte Entwicklung I, Datenkommunikation
Voraussetzungen:	Zulassung zum Praktikum erfolgt bei bestandener Modulprüfung in Programmieren I oder Programmieren II
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden) In diesem Modul findet zusätzlich eine Blockveranstaltung "Wissenschaftliches Arbeiten" statt.
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden kennen die Eigenschaften und den Einsatzbereich von Echtzeitbetriebssystemen. Ihnen ist der OSEK-Standard aus dem Automobilbereich mit seinen Begriffen und einschränkenden Annahmen bekannt. Sie haben praktische Erfahrung mit der Implementierung des OSEK-Standards der Firma Vector Informatik GmbH namens Microsar OS und können mit den zugehörigen Werkzeugen einfache, eingebettete Applikationen konfigurieren und implementieren. Sie sind ferner in der Lage, mit der BOOST C++ Library plattformunabhängige, verteilte Anwendungen zu erstellen.

Die Studierenden sind in der Lage, selbständig wissenschaftliche Arbeiten zu verfassen.

Lehrinhalte:

- Eigenschaften und Komponenten von Echtzeitbetriebssystemen
- Echtzeitanforderungen
- Unterbrechungsbehandlung
- Scheduling-Verfahren
- Synchronisation und Interprozesskommunikation
- plattformunabhängiges Programmieren verteilter Anwendungen mit der BOOST C++ Library
- OSEK-Standard: Task-Konzept, Konformitätsklassen, Prioritäten-Levels, Event, Counter, Alarm, Message, Hook-Funktionen, Konfiguration, OIL, Design Patterns, API
- Erstellen von Anwendungen mit Microsar OS auf einer eingebetteten Hardware
- Methoden zum Verfassen einer wissenschaftlichen Arbeit bzgl. Inhalt, Stil und Form

Literatur:

- M. Homann: OSEK: Betriebssystem-Standard für Automotive und Embedded Systems, mitp-Verlag, 1. Bonn, 2005
- J. T. Benra, W. A. Halang: Software-Entwicklung für Echtzeitsysteme, Springer Verlag., 1. Auflage, Heidelberg, 2009
- A. S. Tanenbaum, H. Bos: Moderne Betriebssysteme, Pearson Studium, 4. Auflage, München, 2016
- W. Stallings: Operating Systems, Pearson, 9. Auflage, 2018
- J. Wietzke, M. Tien Tran: Automotive Embedded Systeme, Springer Verlag, 1. Auflage, Berlin Heidelberg, 2005
- J. Schäuffele, Th. Zurawka: Automotive-Software-Engineering, Vieweg Verlag, 3. Auflage, Wiesbaden, 2006
- M. Kornmeier: Wissenschaftlich schreiben leicht gemacht, utp Verlag, 8. Auflage; 2018

Algorithmen und Datenstrukturen

AIF411

Modulverantwortlicher:	Prof. Andreas Siebert, Ph.D.
Dozent:	Prof. Andreas Siebert, Ph.D.
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im vierten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmierkenntnisse
Voraussetzungen:	-
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden kennen die wesentlichen Datenstrukturen und Algorithmen, die für die Softwareentwicklung benötigt werden. Sie haben die Implementierung von effizienten Algorithmen eingeübt. Sie haben ein Verständnis für die asymptotische Laufzeitkomplexität von Algorithmen entwickelt und können sie analytisch herleiten.

Lehrinhalte:

- Komplexität von Algorithmen, Landau-Symbole, Master-Theorem
- Verantwortungsvolle Algorithmische Systeme
- Paradigmen der Algorithmenentwicklung
- Sortieralgorithmen
- NP-vollständige Probleme
- Randomisierte Algorithmen
- Parallele Algorithmen
- Dynamische Programmierung
- Ausgewählte Algorithmen

Literatur:

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: An Introduction to Algorithms, 4th ed., The MIT Press, 2022.
Tim Roughgarden, Algorithms Illuminated, 2022.

Datenkommunikation

AIF412

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Prof. Dr. Dieter Nazareth
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im dritten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I/II, Modellbasierte Entwicklung I, Software Engineering I, Grundlagen der Informatik
Voraussetzungen:	Zulassung zum Praktikum erfolgt bei bestandener Modulprüfung in Programmieren I oder Programmieren II
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden besitzen fundierte Kenntnisse in den wichtigsten Bustechnologien aus der Fahrzeugtechnik: K-Line, LIN, CAN, FlexRay, MOST, Automotive Ethernet. Sie kennen ferner die wichtigsten Transport- und Diagnoseprotokolle und können mit dem Werkzeug CANoe (Fa. Vektor) an Fahrzeugbussen Messungen durchführen sowie Restbussimulationen aufbauen. Die Studierenden sind damit in der Lage, Fahrzeugfunktionen zu entwickeln, die auf mehrere Steuergeräte verteilt sind.

Lehrinhalte:

- funktions-, domänen- und modulatorientierte Bordnetzarchitekturen
- elektrotechnische Grundlagen und Bustopologien
- Adressierung von Botschaften und Buszugriffsverfahren
- Physical und Data Link Layer von K-Line, CAN, LIN, FlexRay, MOST, Automotive Ethernet
- Protokolle des Transport Layer: ISO-TP, TP 2.0, Flexray TP, UDP, TCP, DoIP
- Protokolle des Application Layer: KWP 2000, UDS, OBD, Some/IP, TSN
- Protokolle für Messen, Kalibrieren und Diagnose: ASAM, CCP, XCP, AML (A2L), FIBEX
- Entwicklungsprozess mit CANoe: Netzwerkdesign und Simulation, Restbussimulation, Integration und Test des Netzwerks

Literatur:

- W. Zimmermann, R. Schmidgall: Bussysteme in der Fahrzeugtechnik, Vieweg+Teubner, 5. Auflage, Wiesbaden, 2014
- Ch. Marscholik, P. Subke: Datenkommunikation im Automobil, Hüthig Verlag, Heidelberg, 2007
- W. Lawrenz: CAN Controller Area Network, Hüthig Verlag, 4. Auflage, Heidelberg, 2000
- A. Grzemba, J. von der Wense: LIN-Bus, Franzis Verlag, 1. Auflage, 2005
- M. Rausch: FlexRay, Hanser Verlag, 1. Auflage, 2007
- G. Schnell, B. Wiedemann (Hrsg.): Bussysteme in der in der Automatisierungs- und Prozesstechnik, Vieweg+Teubner, 7. Auflage, Wiesbaden, 2008

Modellbasierte Entwicklung II

AIF413

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Prof. Dr. Dieter Nazareth
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im vierten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I
Voraussetzungen:	Zulassung zum Praktikum erfolgt bei bestandener Prüfung in Programmieren I oder Programmieren II
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden sind in der Lage, mit Hilfe des Werkzeugs ASCET Developer modellbasiert komplexe Fahrzeugfunktionen zu entwickeln. Sie sind fähig komplexe Modelle unter Verwendung der jeweils adäquaten Modellierungstechnik zu erstellen. Sie können diese Modelle dann in einer simulierten Experimentierumgebung ablaufen lassen. Die Studenten haben den Unterschied zwischen physikalischer Modellierung und Implementierung verstanden und können physikalische Daten durch Festkommaarithmetik implementieren. Die Studenten überblicken den gesamten modellbasierten Entwicklungsprozess mit seinen alternativen Werkzeugen und können ihn in der Praxis anwenden.

Lehrinhalte:

- Erstellen einer Anforderungsspezifikation
- Objektorientierte Modellierung mit Klassen, hierarchischen Zustandsautomaten und ESDL
- Datenfluss- und Kontrollflussdiagramme
- Definition eines Projekts für verschiedene Plattformen
- Aufbau einer Experimentierumgebung
- Durchführung von Offline-Simulationen
- Einführung in das Betriebssystem OSEK
- Verwendung von Kennlinien und Kennfeldern
- Implementierung der Daten mit Festkommaarithmetik

Literatur:

User Manuals und Online Hilfe von ASCET Developer (Fa. ETAS)
J. Schäuffele, Th. Zurawka: Automotive-Software-Engineering, Springer/Vieweg, 6. Auflage

Praktische Zeit im Betrieb

AIF590

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	-
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im fünften Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	-
Voraussetzungen:	Erfolgreiches Bestehen aller Prüfungen der ersten beiden Studiensemester
Leistungspunkte:	22
Arbeitsaufwand:	80 Arbeitstage Präsenzzeit im Betrieb
Lehrformen:	Tätigkeit in der Wirtschaft
Leistungsnachweise und Prüfung:	Praktikumsbericht (Benotung: mit/ohne Erfolg)

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden haben vertiefte Kenntnisse über technische und organisatorische Problemlösungen in Betrieben.

Lehrinhalte:

Die Studierenden werden zum selbständigen und eigenverantwortlichen Arbeiten in praxisrelevanten Automobilinformatikprojekten angeleitet. Die Arbeit sollte möglichst in einem typischen Automobilinformatikprojekt erfolgen.

Literatur:

Tätigkeitsspezifisch

Praxisseminar

AIF591

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Prof. Dr. Dieter Nazareth
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im fünften Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	-
Voraussetzungen:	Erfolgreiches Bestehen aller Prüfungen des ersten Studienabschnitts. AIF590 muss parallel zu AIF591 belegt werden oder bereits abgeleistet sein
Leistungspunkte:	3
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 60 Stunden Selbststudium
Lehrformen:	2 SWS Seminar mit Kurzreferaten und Diskussion
Leistungsnachweise und Prüfung:	Teilnahmepflicht, benoteter Vortrag über das Praktikum AIF590

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden lernen verschiedene Felder der Automobilinformatik in der beruflichen, außeruniversitären Praxis kennen. Sie können ein umfangreiches Projekt verständlich und wohlstrukturiert präsentieren.

Lehrinhalte:

- Erfahrungsaustausch
- Anleitung und Beratung
- Fachliche Diskussion
- Präsentationsstil

Literatur:

Tätigkeitsspezifisch

Softwarearchitekturen

AIF612

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Prof. Dr. Dieter Nazareth
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im vierten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I/II, Modellbasierte Entwicklung I/II, Datenkommunikation
Voraussetzungen:	Zulassung zum Praktikum erfolgt bei bestandener Prüfung in Programmieren I oder Programmieren II
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden besitzen ein tiefes Know-How über die Strukturierung von Software in einem Steuergerät. Sie kennen speziell die AUTOSAR Softwarearchitektur und die AUTOSAR Methodik zur Entwicklung steuergeräteunabhängiger Software.

Lehrinhalte:

- Begriffsdefinition
- Einführung in Komplexität
- Grundbegriffe der Strukturierung
- Prinzipien der Architekturbildung
- Einführung in AUTOSAR
- Die AUTOSAR Methodik
- Die Systemsicht
- Kommunikationsmechanismen
- Steuergeräte- und Netzwerksicht
- AUTOSAR Basissoftware
- UML
- SysML
- Grundlagen Anforderungsmanagement

Literatur:

Olaf Kindel, Mario Friedrich: Softwareentwicklung mit AUTOSAR: Grundlagen, Engineering, Management in der Praxis, dpunkt Verlag
Oliver Alt: Modellbasierte Systementwicklung mit SysML, Carl Hanser Verlag
Jörg Schäuuffele, Thomas Zurawka: Automotive Software Engineering, Springer Verlag
Fabian Wolf: Fahrzeuginformatik, Springer Verlag
Konrad Reif: Automobilelektronik, Springer Verlag
Miroslaw Staron: Automotive Software Architectures, Springer Verlag

Entwicklung sicherheitskritischer Systeme

AIF620

Vorlesung 1: Einführung in das Systems Engineering

Vorlesung 2: Funktionale Sicherheit

Modulverantwortlicher:	Prof. Dr. Sebastian Schröter
Dozent:	Vorlesung 1: Dr. Sebastian Schröter, Vorlesung 2: Dietmar Kinalzyk
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im sechsten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I/II, Modellbasierte Entwicklung I, Datenkommunikation
Voraussetzungen:	Zulassung zum Praktikum erfolgt bei bestandener Modulprüfung in Programmieren I oder Programmieren II
Leistungspunkte:	7
Arbeitsaufwand:	Vorlesung 1: 30 Stunden Präsenzzeit im Unterricht, 40 Stunden Selbststudium Vorlesung 2: 60 Stunden Präsenzzeit im Unterricht, 80 Stunden Selbststudium
Lehrformen:	6 SWS seminaristischer Unterricht mit Übungen und Praktika
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

von „**Einführung in das Systems Engineering**“: Die Studierenden kennen die wesentlichen Prinzipien des Systems Engineering. Sie sind in der Lage, zweckdienliche Methoden und Vorgehensweisen zur strukturierten Problemlösung und Entscheidungsfindung innerhalb eines Produktentstehungsprozesses anzuwenden.

von „**Funktionale Sicherheit**“: Die Studierenden sind in der Lage, einfache sicherheitskritische Systeme zu entwerfen und gegen Fehler abzusichern. Sie kennen die Gefahrenanalyse, Risikobestimmung und die wichtigsten Maßnahmen um das Risiko durch auf das notwendige Niveau abzusenken. Die Studierenden kennen den Umgang mit Definitionen vom Standard ISO26262, den Zusammenhang mit Qualitätsmaßnahmen und den Einfluss auf den Entwicklungsprozess.

Lehrinhalte:

von „Einführung in das Systems Engineering“:

- Ziele, Definitionen und Disziplinen des Systems Engineering
- Generelle Prinzipien des Systems Engineering
- Prinzipien der Strukturierung
- Ausgewählte Themen aus dem Systems Engineering
 - Systemlebenszyklus
 - Vorgehensmodelle
 - Tailoring
 - Bedarfs- und Anforderungsanalyse
 - Schnittstellen
 - Architektur und RFLP Logik
 - Integration, Verifikation und Validierung
 - Konfigurationsmanagement
 - Änderungsmanagement
 - Entscheidungsmanagement
 - Betrieb, Wartung, Entsorgung
- Systemmodellierung im Safety Kontext
- Die Anwendung der erlernten Methoden erfolgt durch Ausarbeitungen von Übungen in Teams.

von „Funktionale Sicherheit“:

- Einführung in die Funktionale Sicherheit, Gefahren, Risiko, Standards und Zielbestimmung
- Sicherheitsziel, sicherer Zustand, Fehlertoleranzzeit
- Zuverlässigkeit, Ausfallrate, Verfügbarkeit
- Fehlermodelle, Fehleranalyse, Minderung der Auswirkung, Metriken
- Hierarchie Ebenen im System und Aufteilung der Fehlerwahrscheinlichkeit
- Funktionales Sicherheitskonzept, Sicherheitsanalysen, Methoden
- Technisches Sicherheitskonzept, Selbstüberwachung, Integrität, Notlauf
- Dekomposition durch Diversität und unabhängige Redundanz
- Ableitung von HW und SW design
- Testmethoden und -verfahren.
- Sicherere Bus- Kommunikation
- Entwicklungsprozesse, Qualität, Audit, Assessment
- Anwendungsbeispiele aus der Praxis

Literatur:

zu „Einführung in das Systems Engineering“:

R. Haberfellner: Systems Engineering – Grundlagen und Anwendung, Orell Füssli Verlag, Zürich, 2015.
D.D. Walden, et. al. (Hrsg.): Systems engineering handbook : a guide for system life cycle processes and activities / prepared by International Council on Systems Engineering (INCOSE) – 4th edition.
SEBoK Editorial Board. 2021. The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.5, R.J. Cloutier (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens Institute of Technology, www.sebokwiki.org.

zu „Funktionale Sicherheit“:

P. Löw, R. Pabst, E. Petry: Funktionale Sicherheit in der Praxis: Anwendung der DIN EN 61508 und ISO/DIS 26262 bei der Entwicklung von Serienprodukten, dpunkt Verlag, 1. Auflage, 2010
H.-L. Ross: Funktionale Sicherheit im Automobil, Hanser Verlag, 2014
D. Dürholz, S. Herrmann, R. Stärk: Safety Essentials, Kugler Maag Verlag, 2014
Gebhardt V., Rieger G., Mottok J., C. Gießelbach: Funktionale Sicherheit nach ISO 26262, dpunkt Verlag

Autonome Fahrzeuge

AIF630

Modulverantwortlicher:	Prof. Dr. Martin Pellkofer
Dozent:	Prof. Dr. Martin Pellkofer
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im sechsten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Programmieren I (Programmierkenntnisse in C/C++), Modellbasierte Entwicklung I (Grundkenntnisse in Matlab/Simulink)
Voraussetzungen:	
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden kennen den Stand der Technik bei hoch automatisierten und autonom fahrenden Landfahrzeugen. Dies beinhaltet die eingesetzte Sensorik, Sensordatenverarbeitung, Sensordatenfusion, Fahrzeugregelung, Navigation und Planung, sowie Systemdesign und Test. Die Studierenden haben sich ferner mit den ethischen und rechtlichen Fragen auseinandergesetzt, welche autonome Fahrzeuge aufwerfen.

Lehrinhalte:

- Historischer und thematischer Überblick
- Sensoren autonomer Fahrzeuge: Kamera, Lidar, Radar
- Paradigmen beim Autonomen Fahren: Modulare Pipelines, End-to-End-Learning, Direct Perception
- MLP, CNN, Imitation Learning, Direct Perception
- Odometrie, SLAM, Lokalisierung
- Straßen- und Fahrspurerkennung
- Stereoskopisches Sehen, Freiraumerkennung, Optischer Fluss
- Objektdetektion: Detektionsleistung, Region-Based-CNN, Fast(er) R-CNN
- Objektverfolgung: Single & Multi Object Tracking, Assoziationsproblem, Track-Level-Fusion
- Sensordatenfusion und Zustandsschätzung: Bayes-Filter, (Erweiterter) Kalman-Filter
- Routenplanung, Verhaltensplanung, Trajektorienplanung
- Ethische und rechtliche Fragen beim autonomen Fahren

Literatur:

- H. Winner, S. Hakuli, F. Lotz, C. Singer: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer, 2015
E. D. Dickmanns: Dynamic Vision for Perception and Control of Motion, Springer, 2007
M. Maurer, J. Ch. Gerdes, B. Lenz, H. Winner (Hrsg.): Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte, Springer, 2015
M. Botsch, W. Utschick: Fahrzeugsicherheit und autonomes Fahren, Hanser, 2020
Q. Zhou, Z. Shen, B. Yong, R. Zhao, P. Zhi: Theories and Practices of Self-Driving Vehicles, Elsevier, 2022
M. Du: Autonomous Vehicle Technology, Springer, 2023

Fachbezogenes Wahlpflichtmodul

AIF670

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Dozenten der Hochschule Landshut
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im sechsten oder siebten Studiensemester. Vor Beginn des sechsten Studiensemesters wird eine Liste der angebotenen Fächer mit ihren Beschreibungen veröffentlicht.
Dauer:	ein Semester
Vorkenntnisse:	Erster Studienabschnitt oder vergleichbare Kenntnisse
Voraussetzungen:	siehe individuellen Modulbeschreibungen
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit im Unterricht 90 Stunden Selbststudium
Lehrformen:	Modulspezifisch
Leistungsnachweise und Prüfung:	Leistungsnachweise und Prüfungen werden in den individuellen Modulbeschreibungen festgelegt.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden sind mit ausgewählten, fachbezogenen Wissensgebieten oder erweiterten Fertigkeiten in speziellen Anwendungen, die der individuellen Vorbereitung auf die berufliche Praxis dienen, vertraut.

Lehrinhalte:

Siehe individuelle Fachbeschreibungen

Literatur:

Siehe individuelle Fachbeschreibungen

Prozessrechentechnik

AIF710

Modulverantwortlicher:	Prof. Dr. Martin Pellkofer
Dozent:	Prof. Dr. Martin Pellkofer
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im siebten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Erster Studienabschnitt oder vergleichbare Kenntnisse
Voraussetzungen:	Zulassung zum Praktikum erfolgt bei bestandener Modulprüfung in Programmieren I oder Programmieren II
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden kennen die Eigenschaften und Charakteristika, in denen sich eingebettete Systeme von IT-Systemen unterscheiden. Sie verfügen über das notwendige Wissen in Systemdynamik und Regelungstechnik, um Prozessrechner als digitale Filter und Regler einsetzen zu können. Die Studierenden kennen neben Prozessoren weitere Arten von Verarbeitungseinheiten mit ihren Vor- und Nachteilen. Die Studierenden können externe Geräte wie Sensoren oder Aktoren an einem eingebetteten Rechner anbinden.

Lehrinhalte:

- Unterschiede zwischen eingebetteten Systemen und IT-Systemen
- Aufbau eingebetteter Systeme
- Grundlagen in Systemdynamik und Regelungstechnik
- Realisierung einfacher digitaler Filter und Regler mit Prozessrechner
- Sensoren und Aktuatoren eingebetteter Systeme
- Analoge und digitale I/O, Pulsweitenmodulation
- Analog/Digital-Wandlung, Alias-Effekte, Nyquist-Shannon-Abtasttheorem
- Modellbasiertes Design: Motivation, Grundprinzipien, x-in-the-Loop-Simulationen
- Entwicklung eingebetteter Softwaresysteme: Optimierung hinsichtlich Ressourcenverbrauch, Zuverlässigkeit, Risiko, Funktionale Sicherheit, Reliability Engineering
- Programmierung eines Mikrocontrollers (MPC560*B von NXP): Timed I/O, Interrupts, ADC, DMA
- Entwicklung einer kamerabasierten Querführung für ein Modellfahrzeug

Literatur:

- P. Marwedel: Eingebettete Systeme, Springer Verlag, Heidelberg, 2008
K. Berns, B. Schürmann, M. Trapp: Eingebettete Systeme, Vieweg+Teubner, 1. Auflage, Wiesbaden, 2010
T. Benra, W. A. Halang: Software-Entwicklung für Echtzeitsysteme, Springer, Heidelberg, 2009
K. Berns, B. Schürmann, M. Trapp: Eingebettete Systeme, Vieweg+Teubner, 1. Auflage, Wiesbaden, 2010
Handbücher der benutzten Hardware und Software

Seminar

AIF720

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Dozenten der Fakultät Informatik
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch
Angebot:	im siebten Studiensemester
Dauer:	ein Semester
Vorkenntnisse:	Informatik-Kenntnisse aus den ersten sechs Semestern des Bachelor-Studiums oder vergleichbare Kenntnisse
Voraussetzungen:	-
Leistungspunkte:	3
Arbeitsaufwand:	30 Stunden Präsenzzeit 60 Stunden Selbststudium
Lehrformen:	Fachliche Präsentationen durch die Studierenden und anschließende Diskussionen
Leistungsnachweise und Prüfung:	Teilnahmepflicht, benotete Präsentationen.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden sind in der Lage sich ein komplexes fachliches Thema aus der Literatur selbstständig zu erarbeiten. Sie können das Thema in einem fachlichen Vortrag unter Zuhilfenahme moderner Medien präsentieren und mit einem technisch versierten Publikum eine Diskussion über die Präsentationsinhalte führen.

Lehrinhalte:

aktuelle Themen der Automobilinformatik

Literatur:

aktuelle Themen der Automobilinformatik

Bachelor-Arbeit

AIF790

Modulverantwortlicher:	Prof. Dr. Dieter Nazareth
Dozent:	Dozenten der Hochschule Landshut. Mindestens einer der Prüfer ist ein hauptamtlicher Professor der Fakultät Informatik
Studiengang:	Bachelor
Modultyp:	Pflichtfach
Sprache:	Deutsch oder Englisch
Angebot:	Nach Ableistung des praktischen Semesters
Dauer:	5 Monate
Vorkenntnisse:	-
Voraussetzungen:	Erfolgreiche Ableistung der praktischen Zeit im Betrieb (Modul AIF590).
Leistungspunkte:	12
Arbeitsaufwand:	360 Stunden selbstständige Arbeit
Lehrformen:	Selbstständiges Arbeiten
Leistungsnachweise und Prüfung:	Schriftliche Bachelor-Arbeit, Kolloquium.

Qualifikationsziele und Inhalte:**Qualifikationsziele:**

Die Studierenden können ein etwas größeres, aber zeitlich klar begrenztes, praxisbezogenes Automobilinformatik-Thema eigenständig und wissenschaftlich bearbeiten. Sie sind in der Lage, Problemstellungen und deren Lösungen schriftlich darzustellen und mündlich zu präsentieren.

Lehrinhalte:

Abhängig vom Thema der Arbeit

Literatur:

Abhängig vom Thema der Arbeit

Module aus dem Bachelor-Studiengang Automobilwirtschaft und -technik

Einige Module des Studiengangs Automobilinformatik werden aus dem Bachelor-Studiengang „Automobilwirtschaft und -technik“ der Fakultät „Elektrotechnik und Wirtschaftsingenieurwesen“ der Hochschule Landshut importiert (Importstudiengang). Die Inhalte dieser Module werden durch das vorliegende Modulhandbuch nicht festgelegt, sondern es gilt das Modulhandbuch des Importstudiengangs in seiner aktuellen Version. Insbesondere sind die modulspezifischen Teilnahmevoraussetzungen zu beachten.

Untenstehende Tabelle fasst die aus dem Bachelor-Studiengang „Automobilwirtschaft und -technik“ stammenden Module des Studiengangs Automobilinformatik (Importmodule) zusammen.

Module des Studiengangs Automobilinformatik		Importstudiengang	Fakultät der Hochschule Landshut	Modulnummer
AIF120	Ingenieurmathematik I	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T110 ¹
AIF140	Grundlagen der Elektrotechnik	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T120
AIF150	Technische Mechanik	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T140
AIF220	Ingenieurmathematik II	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T210 ²
AIF240	Elektronik und Messtechnik	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T220
AIF241	Angewandte Physik	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T240
AIF340	Regelungstechnik	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T320
AIF350	Konstruktion und Entwicklung	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T311 ³
AIF450	Grundlagen der Automobiltechnik	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T410
AIF650	Fahrwerktechnik	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T610 ⁴
AIF651	Antriebskonzepte	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T620 ⁵
AIF750	Karosserietechnik	Bachelor „Automobilwirtschaft und -technik“	Elektrotechnik und Wirtschaftsingenieurwesen	T640 ⁶

¹Bei Automobilinformatik 7 ECTS

²ohne Teilnahme am Vorlesungsteil „Statistik“

³ohne Teilnahme am Praktikum

⁴Name des Moduls: Automobiltechnik I: Fahrwerk

⁵Name des Moduls: Automobiltechnik II: Antriebskonzepte

⁶Name des Moduls: Automobiltechnik IV: Karosserietechnik