

Anlage 2 zur StO Master Embedded Systems vom 4. 12. 2007

Modulhandbuch für den Master-Studiengang Embedded Systems
(Ansprechpartner für das Modulhandbuch: Studienfachberater Prof. Keutner,
Email: Keu@gmx.de)

Die im Modulhandbuch angegebenen Anteile unterschiedlicher Prüfungsformen zur Bildung der Modulnote stellen einen Vorschlag dar. Das exakte Verhältnis der Prüfungsergebnisse an der Modulnote wird den Studierenden innerhalb der Belegungszeit von der jeweiligen Lehrkraft nachvollziehbar / schriftlich (auch im Intranet) mitgeteilt.

Modulnummer	Modulname	Koordinator/in
ACS	Vertiefung Regelungstechnik	Prof. Ottens
AMS	Autonome mobile Systeme	Prof. Dr. Sommer
ARS	Vertiefung Echtzeitsysteme	Prof. Dr. Buchholz
AW	Allgemeinwissenschaftliches Ergänzungen	Dekan/in FB I
CES	Konfigurierbare eingebettete Systeme	Prof. Dr. Teppner
CPE	Computational Engineering	Prof. Ottens
CTE	Modul mit aktuellen Inhalten zu Embedded Systems	Prof. Keutner
ESD	Entwurf eingebetteter Systeme	Prof. Dr. Rozek
MAV	Maschinelles Sehen	Prof. Dr. Buchholz
MBD	Modellbasierter Entwurf	Prof. Dr. Rozek
MTH	Masterarbeit	Prof. Keutner
NTC	Netzwerk-Programmierung	Prof. Dr. Görlich
OEX	Mündliche Prüfung	Prof. Ottens
RCP	Schneller Regler-Prototypenentwurf	Prof. Ottens
RQE	Anforderungsmanagement	Prof. Gramm

Modulnummer	MAV
Titel	Maschinelles Sehen / Machine Vision
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	Die Studierenden können die für eine Problemstellung des maschinellen Sehens benötigten Komponenten sachgerecht auswählen (Beleuchtung, Kamera, Objektive, Bus-Anbindung). Sie sind außerdem in der Lage, unter Verwendung einer graphisch orientierten Entwicklungsumgebung Bilder aufzubereiten, diese zu segmentieren, Merkmale zu extrahieren und eine Klassifizierung durchzuführen. Sie können weiterhin ein typisches Problem der digitalen Bildverarbeitung unter den Randbedingungen eines Embedded Systems bei eigener Erweiterung einer bestehenden Funktionsbibliothek implementieren.
Voraussetzungen	keine
Niveaustufe	1. Studienplansemester
Lernform	Seminaristischer Unterricht und Übungen am Digitalrechner
Status	Pflichtmodul
Häufigkeit des Angebotes	Sommersemester
Prüfungsform	Klausur. Voraussetzung zur Teilnahme sind erfolgreich gelöste Übungsaufgaben
Ermittlung der Modulnote	Klausur
Anerkannte Module	Module vergleichbaren Inhalts

Inhalte	<p><u>Im seminaristischen Unterricht</u></p> <ul style="list-style-type: none">• Komponenten von Bildverarbeitungssystemen (Beleuchtung, Objektive, Kameras,• Frame-Grabber, Software• Punktoperationen• lokale Operatoren• globale Operatoren• Bereichssegmentierung• Kontursegmentierung• Hough-Transformation• morphologische Bildverarbeitung• Mustererkennung• Besonderheiten beim Einsatz in Embedded Systems <p><u>In den Übungen</u></p> <p>Am Digitalrechner wird in Gruppen von zwei Studierenden ein komplexeres Projekt unter Verwendung einer kommerziellen Entwicklungsumgebung realisiert, um exemplarisch den vermittelten Stoff zu vertiefen. In einem zweiten Projekt wird auf dem im Modul AMS verwendeten embedded system ein Problem der Echtzeit-Farbbildverarbeitung realisiert.</p>
Literatur	<p>Bässmann, Kreyss: „Bildverarbeitung Ad Oculos“, Springer</p> <p>Burger, Burge: „Digitale Bildverarbeitung : eine Einführung mit Java und ImageJ“, 2. Aufl., Springer, 2006</p> <p>Unterlagen zum Entwicklungssystem Halcon (pdf-Dateien auf den Rechnern im Labor)</p>
Weitere Hinweise	<p>Dieses Modul wird auf Deutsch angeboten</p>

Modulnummer	RCP
Titel	Schneller Regler-Prototypen-Entwurf / Rapid Control Prototyping
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	<p>Neu entwickelte digitale Filter- oder Regelalgorithmen werden nach durchgeführten Tests in der Entwicklungsumgebung (i.a. einem PC) mittels C oder Assembler auf dem Zielsystem, z.B. einem eingebetteten Mikrocontroller, implementiert. Dabei sind die Entwicklungs- und die Implementierungsumgebung häufig zwei völlig verschiedene Arbeitsumgebungen und die Programmerstellung zur Implementierung ist sehr zeitaufwendig. Dies ist insbesondere der Fall, wenn der Test offline durchgeführt wurden, die Implementierung aber in Echtzeit arbeiten soll. Ähnliche Probleme ergeben sich, wenn die Tests mit einem Gleitkommaprozessor durchgeführt wurden, die Implementierung aber nur über einen Festkommaprozessor verfügt. Die Folge ist eine Vermischung der Probleme des Algorithmenentwurfs und der Implementierung.</p> <p>Vom Programmsystem Matlab/Simulink wird z.B. eine einheitliche Entwicklungs- und Implementierungsplattform, der sogenannte "Realtime Workshop" zur Verfügung gestellt, der quasi auf Knopfdruck einen in einer einfachen grafischen Arbeitsumgebung entwickelten Algorithmus in eine Echtzeit-Anwendung für ein gewähltes Zielsystem, z.B. einen Windows-PC oder ein eingebettetes System (Mikrocontroller, Signalprozessor), compiliert und implementiert. Dieser Vorgang wird in der Literatur häufig mit "Rapid Prototyping" zum Regler- und Filterentwurf bezeichnet.</p> <p>Basierend auf ihren Kenntnissen und Fertigkeiten auf den Gebieten Systemtheorie, Regelungstechnik und Matlab / Simulink erwerben die Studierenden Kenntnisse der Entwicklungsphilosophie sowie der Bedeutung und Notwendigkeit der einzelnen Entwicklungsschritte zum Filter- und Reglerentwurf. Sie besitzen die Fertigkeit, dieses weltweit verbreitete Entwicklungswerkzeug umfassend zu nutzen, und haben die Kompetenz, den erheblichen Gewinn an Entwicklungszuverlässigkeit und Produktivitätsfortschritt zu erkennen und zu beurteilen.</p>
Voraussetzungen	Empfehlung: Gute Systemtheorie-, Regelungstechnik- und Matlab/Simulink-Kenntnisse
Niveaustufe	1. Studienplansemester
Lernform	Seminaristischer Unterricht mit Laborübungen
Status	Pflichtmodul
Häufigkeit des Angebotes	Sommersemester
Prüfungsform	Klausur und Laborbericht mit Rücksprache
Ermittlung der Modulnote	70% Klausurnote, 30% Laborbericht und Rücksprache
Anerkannte Module	Module vergleichbaren Inhalts

Inhalte	<p><u>Im seminaristischen Unterricht</u></p> <p>Grundlagen des Rapid Control Prototyping</p> <ul style="list-style-type: none"> • Begriffsbildung, • klassischer und Rapid Prototyping Entwurf, • Hardware-in-the-Loop, • Software-in-the-Loop, • Modellbasierter Entwurf, • Codegenerierung, • Echtzeitbetrieb, • Festkomma- / Gleitkommaprozessor, • marktverfügbare Entwicklungsumgebungen und unterschiedliche Rapid Prototyping – Methoden beim Entwurf von Regelkreisen <p>Anwendung des Rapid Control Prototyping</p> <ul style="list-style-type: none"> • Messtechnische Identifikation der Steuer- und Störverhaltens der Regelstrecke, • Simulation der identifizierten Regelstrecke, • Reglerentwurf, • Simulation des Regelkreises, • Nachoptimierung, • Implementierung des Reglers in das Zielsystem, Echtzeitbetrieb des Regelkreises. <p><u>In den Übungen</u></p> <p>An verschiedenen praktisch realisierten Regelstrecken-Modellen (Temperatur, Drehzahl-, Spannungs-, -Regelstrecken) werden mittels Simulink und dem Realtime-Workshop von Matlab durch Rapid-Prototyping-Methoden</p> <ul style="list-style-type: none"> • das Steuer- und Störverhalten der Regelstrecke experimentell identifiziert, • das Steuer- und Störverhalten der Strecke simuliert, • der Regler entworfen und optimiert, • der Regelkreis simuliert und der Regler in die reale Strecke implementiert und auf der Zielhardware erprobt..
Literatur	Dirk Abel, Alexander Bolling: Rapid Control Prototyping, Springer Verlag
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Modulnummer	ESD
Titel	Entwurf eingebetteter Systeme / Embedded System Design
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	<p>Die Studierenden erlangen vertiefte fachliche und fachübergreifende Kenntnisse im Entwurf, in der Entwicklung und in der Inbetriebnahme von Mikrocontrollersystemen.</p> <p>Sie lernen, wie vorgefertigte Hardware-Funktionsmodule für MSR-Anwendungen eingesetzt, diese miteinander kombiniert und damit zu kompletten Embedded Systemen zusammengefügt und getestet werden.</p> <p>Durch die vertiefende Diskussion moderner Entwicklungsmethoden und neuer Technologien, auch unter dem Aspekt umwelttechnischer Verantwortung, werden sie in die Lage versetzt, Embedded Systeme nicht nur funktional sondern auch zeiteffizient, störungsresistent und zukunftssicher zu entwickeln.</p>
Voraussetzungen	Empfehlung: Mikroprozessorkenntnisse
Niveaustufe	1. Studienplansemester
Lernform	Seminaristischer Unterricht und Übungen an einem Zielsystem (Target)
Status	Pflichtmodul
Häufigkeit des Angebotes	Sommersemester
Prüfungsform	Klausur. Voraussetzung zur Teilnahme sind erfolgreich gelöste Laborübungen
Ermittlung der Modulnote	Klausur
Anerkannte Module	Module vergleichbaren Inhalts

Inhalte	<p><u>Im seminaristischen Unterricht</u></p> <ul style="list-style-type: none"> • Funktion und Architektur von Mikrocontrollern • Vergleich von Prozessorfamilien bzw. -arten und ihren speziellen Eigenschaften (Mikroprozessoren, Mikrocontroller, DSP) • Grundlagen und Systemaufbau von Embedded-Systemen mit verschiedenen Mikrocontrollern inklusive Peripherieanbindung, z.B. A/D- und D/A-Wandlung • Funktionserläuterungen komplexer Controller-Bausteine: z.B. MMU-, DMA-, Grafik-, Disk- und Ethernet-Controller • Entwurf von Peripheriekarten • Systementwicklung mit Hilfe von Standard Bus-Systemen: z.B. PCI-Bus, PCI-Express, VME-Bus,... • Serielle Bussysteme: z.B. I2C, USB,... • Systemprojektierung mit Hilfe von Mezzanine-Boards • Testumgebung am Beispiel von Boundary Scan • Systemintegration: Programmieren, Debuggen, Testen sowie Inbetriebnahme von Embedded Systemen • Aufbau von Mehrprozessorsystemen • Applikations- und Schaltungsbeispiele
	<p><u>In den Übungen</u></p> <ul style="list-style-type: none"> • Einführung in ein professionelles Mikro-Controller Zielsystem (Kompetenz) • Einführung und Nutzung einer professionellen Entwicklungsumgebung (Kompetenz) • Inbetriebnahme eines vorgegebenen Zielsystems zur Lösung praktischer Aufgabenstellungen aus dem MSR-Bereich (Fachkompetenz) • Entwurf und Realisierung eigener praxisnaher Peripherieschaltungen • Inbetriebnahme des Gesamtsystems (Fachkompetenz)
Literatur	<p>Rainer Bermbach: Embedded Controller; Hanser</p> <p>Schmitt, v. Wendorff, Westerholz: Embedded-Control-Architekturen; Hanser</p> <p>Sikora, Drechsler: Software-Engineering und Hardware-Design; Hanser</p>
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Modulnummer	NTC
Titel	Netzwerkprogrammierung / Network Computing
Credits	5 Cr
Präsenzzeit	2 SWS
Workload	Seminar: ~ 36 h Selbstlernzeit: ~ 114 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	Die Studierenden entwickeln selbständig nach eigenständigen Literaturrecherchen eine Anwendung in einem der Gebiete der Inhaltsliste. Sie erstellen eine wissenschaftliche Darstellung und Präsentation der benutzten Technologien und der entwickelten Anwendung. Aus diesen Tätigkeiten resultierend besitzen die Studierenden Fähigkeiten zum Entwurf, zur Entwicklung und Administration Verteilter Systeme auf Basis aktueller Architekturen und Technologien. Sie beherrschen wissenschaftliche Methodik und Fähigkeiten und Fertigkeiten zur praktischen Umsetzung, Dokumentation und Präsentation.
Voraussetzungen	keine
Niveaustufe	1. Studienplansemester
Lernform	Seminar
Status	Pflichtmodul
Häufigkeit des Angebotes	Sommersemester
Prüfungsform	Dokumentation und Präsentation einer selbst entwickelten Verteilten Anwendung im Seminar
Ermittlung der Modulnote	Bewertung der Anwendung und der Präsentation
Anerkannte Module	Module vergleichbaren Inhalts
Inhalte	Ziele und Formen wissenschaftlicher Veröffentlichungen und Präsentationen Themen aus Entwurf und Programmierung Verteilter Systeme, z.B.: <ul style="list-style-type: none"> • Client/Server- und dienstorientierte Architekturen, Grid Computing • Middleware (prozedur-, objekt-, Web- oder Komponentenbasiert), Verteilungstransparenz • Routingstrategien, Verfahren zur Wegebestimmung • Sicherheit: Kryptographie, Zertifizierung, PKI, Firewalls, Intrusion Detection • aktuelle Protokolle der Anwendungsebene • Netzwerkmanagement • Embedded Microcontroller, Sensornetzwerke
Literatur	Gemäß den gewählten Themen.
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Modulnummer	ARS
Titel	Vertiefung Echtzeitsysteme / Advanced Realtime Systems
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	<p>Die Studierenden lernen durch die Realisierung eines komplexen Praxisprojekts den Umgang mit den benötigten Hard- und Softwarekomponenten kennen. Dazu wird die exemplarische Erstellung eines Gerätetreibers, die Verteilung einer aus mehreren Prozessen bestehenden Applikation auf vernetzte heterogene Embedded Systems und die Erstellung einer geeigneten Visualisierungskomponente benötigt. Ein weiteres wesentliches Lernziel stellen die durchzuführende Projektleitung in der jeweiligen Arbeitsgruppe sowie die Durchführung einer geeigneten Projekt-Präsentation dar.</p> <p>Dadurch sind die Studierenden in der Lage, komplexe Echtzeitsysteme zu analysieren und zu realisieren. Dabei können sie aus verschiedenen Lösungsmöglichkeiten die jeweils geeignete identifizieren, Komponenten mit „harten“ und „weichen“ Echtzeitanforderungen unterscheiden sowie für jede Problemstellung eine geeignete Programmiersprache auswählen. Sie werden in der Lage sein, Projekte zu planen und zu leiten sowie deren Ergebnisse zu präsentieren.</p>
Voraussetzungen	keine
Niveaustufe	1. Studienplansemester
Lernform	Seminaristischer Unterricht und Übungen am Digitalrechner sowie praktische Übungen im Labor
Status	Pflichtmodul
Häufigkeit des Angebotes	Sommersemester
Prüfungsform	Klausur. Voraussetzung zur Teilnahme sind erfolgreich gelöste Übungsaufgaben sowie erfolgreiche Rücksprachen zu den Laborübungen
Ermittlung der Modulnote	Klausur
Anerkannte Module	Module vergleichbaren Inhalts

Inhalte	<p><u>Im seminaristischen Unterricht</u></p> <ul style="list-style-type: none">• Design von Echtzeitsystemen, Communicating Sequential Processes• Warteschlangen-Analyse• Fehlertoleranz und Ausfallsicherheit• Debugging von verteilten Echtzeitsystemen• Konzepte von Gerätetreibern unter LINUX, RT-LINUX und QNX• Analyse verschiedener Echtzeit-Projekte• Behandlung der bei den zu realisierenden Projekten auftretenden Problemfelder <p><u>In den Übungen</u></p> <p>Am Digitalrechner sind in Gruppen von zwei bis drei Studierenden ca. zwei komplexere Programmsysteme zu entwickeln, die exemplarisch den vermittelten Stoff vertiefen.</p>
Literatur	<ul style="list-style-type: none">• Dokumentation zu QNX-Neutrino: www.qnx.com/developers/docs/6.3.2/neutrino/user_guide• Dokumentation zu RTAI-LINUX: RTAI-Dokumentation RTAI-API-Dokumentation (beides unter: www.rtai.org/documentation) <p>J. Quade, E.-V. Kunst: „Linux-Treiber entwickeln“, dpunkt.verlag</p>
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten.

Modulnummer	AW
Titel	Allgemeinwissenschaftliches Ergänzungen / Obligatory Option General Studies
Credits	5 Cr
Präsenzzeit	4 SWS oder 2+2 SWS
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Allgemeinwissenschaftliche Ergänzungen
Lernziele / Kompetenzen	Die fachübergreifenden Lehrinhalte dienen der interdisziplinären Erweiterung des Fachstudiums und dem Erkennen von Zusammenhängen zwischen Gesellschaft und ihren Teilsystemen, wie z. B. Technik, Wirtschaft, Politik und Recht, unter besonderer Berücksichtigung genderspezifischer Fragestellungen.
Voraussetzungen	keine (Ausnahmen können für die Fremdsprachen festgelegt werden)
Niveaustufe	1. Studienplansemester
Lernform	Seminaristischer Unterricht, Übungen, Referate, Rollenspiele, Textarbeit,
Status	Wahlpflichtmodul
Häufigkeit des Angebotes	Sommersemester
Prüfungsform	Die Prüfungsform wird in der Beschreibung der konkreten Lehrveranstaltungen festgelegt
Ermittlung der Modulnote	Die Ermittlung der Modulnote, gegebenenfalls bei Teilung des Moduls in zwei Teilleistungen für die beiden Teilleistungsnachweise, wird in der Beschreibung der Lehrveranstaltungen festgelegt. Die Modulnote ergibt sich bei Teilung aus dem Mittel (50%/50%) der Leistungsnachweise der beiden Lehrveranstaltungen
Anerkannte Module	Module vergleichbaren Inhalts
Inhalte	Die Lehrinhalte kommen aus den Bereichen (bei Natur- und Ingenieurwissenschaftlichen Studiengängen) Politik und Sozialwissenschaften Geisteswissenschaften Wirtschafts-, Rechts- und Arbeitswissenschaften Fremdsprachen ODER (bei wirtschaftswissenschaftlichen Studiengängen) Politik und Sozialwissenschaften Geisteswissenschaften Natur- und Ingenieurwissenschaften (Module aus Studiengängen der FB II - VIII) Fremdsprachen Bevorzugte Veranstaltungsform ist das Seminar mit studentischen Eigenbeiträgen, damit zugleich die Kommunikations- und Diskussionsfähigkeit geschult wird. Die semesterweise aktualisierten Inhalte sind strukturiert und detailliert beschrieben unter der URL: http://www.tfh-berlin.de/FBI/AW
Literatur	Wird in den jeweiligen Beschreibungen der Lehrveranstaltungen angegeben

Weitere Hinweise

Die Auswahl der Lehrveranstaltungen dieses Moduls obliegt der Eigenverantwortung der Studierenden. Die Auswahl der Lehrveranstaltungen müssen die Studierenden aus den für ihren Studiengang zugelassenen Bereichen treffen (siehe Inhalt)

Modulnummer	CPE
Titel	Computational Engineering
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachübergreifende Vertiefung
Lernziele / Kompetenzen	<p>Aufbauend auf den erworbenen Kenntnissen in Mathematik und deren Anwendung in der Technischen Informatik erarbeiten die Studierenden Kenntnisse über die Grundlagen, die Hintergründe und die Anwendung moderner computergestützter mathematischer Methoden für die Ingenieurwissenschaften::</p> <ul style="list-style-type: none"> • Symbolische Mathematik, • Optimierung und • Neuronale Netze. <p>Die symbolische Mathematik erlaubt die nicht numerische, rechnergestützte allgemeine Lösung mathematischer Gleichungen . Die Methoden der Optimierung erlauben die Lösung von Problemen , die entweder nicht analytisch lösbar sind oder bei denen der Lösungsaufwand zu groß ist. Die Optimierungsverfahren lösen die gestellten Probleme quasi mittels „systematischen Probierens“. Die Methoden der neuronalen Netze ahmen zur Problemlösung die Funktion gekoppelter Nervenzellen nach. Diese sehr moderne Methode ist noch Gegenstand umfangreicher Forschung.</p> <p>Durch rechnergestützte Anwendungen der drei mathematischen Methoden (mit dem Programmsystem Matlab / Simulink) erwerben die Studierenden die Fertigkeit, die Einsatzgebiete der Methoden abschätzen, sie zur Problemlösung anwenden zu können und sich bei Bedarf tiefer in die komplexen Themengebiete einarbeiten zu können.</p>
Voraussetzungen	Empfehlung: Gute Matlab-Kenntnisse
Niveaustufe	2. Studienplansemester
Lernform	Seminaristischer Unterricht mit Hausübung, Übung
Status	Pflichtmodul
Häufigkeit des Angebotes	Wintersemester
Prüfungsform	Klausur. Voraussetzung zur Zulassung zur Klausur: Vorlage aller Hausübungen der Gruppe und Rücksprache „m.E.“
Ermittlung der Modulnote	Klausurnote
Anerkannte Module	Module vergleichbaren Inhalts

Inhalte	<p><u>Im seminaristischen Unterricht</u></p> <ul style="list-style-type: none"> • Symbolische Mathematik: Symbolisches Rechnen auf Digitalrechnern, Abgrenzung zu numerischen Methoden, Beispiele aus linearer Algebra und Analysis, Vereinfachung von Ausdrücken. • Optimierungsmethoden: Einsatzgebiete von Optimierungsmethoden, Formulierung der Zielfunktion, Uni- und Multimodalität, Probleme mit Randbedingungen, Methoden mit und ohne Anforderungen an die Zielfunktion, globale und lokale Methoden • Neuronale Netze: Was sind Neuronale Netze, wo werden Neuronale Netze eingesetzt, Erstellung, Training und Simulation Neuronaler Netze. <p><u>In den Übungen</u></p> <ul style="list-style-type: none"> • Einführung der entsprechenden Toolboxen des CAE-Programms MATLAB / SIMULINK für die obigen Problemstellungen: (Symbolic-, Optimization- und Neural Network-Toolbox) • Lösung einfacher, beispielhafter praktischer Problemstellungen aus den obigen Wissensgebieten mittels MATLAB / SIMULINK.
Literatur	<p>Symbolische Mathematik: Hans Benker: Mathematik mit Matlab, Springer Verlag</p> <p>Optimierungsmethoden: Narayan S. Rau: Optimization Principles, Wiley-IEEE Verlag</p> <p>Neuronale Netze: Andreas Zell: Simulation neuronaler Netze, R. Oldenbourg Verlag</p>
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Modulnummer	MBD
Titel	Modellbasierter Entwurf / Model Based Design
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachübergreifende Vertiefung
Lernziele / Kompetenzen	<p>Der modellbasierte Ansatz vereinfacht den Entwicklungsprozess von der Spezifizierung des Systems über die Modellierung und Simulation bis hin zum Testen und zur Implementierung der gewünschten Anwendung. Die Studierenden erlangen vertiefte und fachübergreifende Kenntnisse im methodischen Entwurf, in der zielgerichteten Systemsimulation und der Funktionsverifikation von Embedded Systemen.</p> <p>Mit Hilfe einer automatischen Generierung von Programm-Code in Produktionsqualität kann auf die zeitaufwändige und fehleranfällige Codierung per Hand verzichtet werden.</p> <p>Durch den modellbasierten Ansatz und die Simulation erhält man sehr früh im Entwicklungsprozess wertvolle Erkenntnisse über das System und vermeidet so Fehlentwicklungen. Spätere Modifikationen oder Anpassungen an geänderte Anforderungen lassen sich auf dieser Grundlage sehr einfach vornehmen. Es lässt sich auch problemlos bestehender Programmcode einbinden, auf beliebige Datenquellen zugreifen und sogar eigene Programm-Module erstellen oder die Codegenerierung auf persönliche Bedürfnisse und individuelle Zielhardware anpassen. In der Summe werden Entwicklungszeit und –kosten eingespart.</p> <p>Durch praktische Übungen an einem MBD-System wird das Verständnis hinsichtlich eines „Rapid Prototyping“ mit Hilfe eines PC-in-the-Loop vertieft und damit die wissenschaftliche und technische Kompetenz der Studierenden gestärkt.</p> <p>Anwender des modellbasierten Ansatzes finden sich in verschiedenen Industriezweigen: Automobil, Luft- & Raumfahrt, Telekommunikation, Verfahrenstechnik, Maschinen- und Anlagenbau und Robotik.</p>
Voraussetzungen	Empfehlung: Mikroprozessor- und Mikrocontrollerkenntnisse
Niveaustufe	2. Studienplansemester
Lernform	Seminaristischer Unterricht und Laborübungen an professionellen Entwicklungs- und Zielsystemen
Status	Pflichtmodul
Häufigkeit des Angebotes	Wintersemester
Prüfungsform	Klausur. Voraussetzung zur Teilnahme sind erfolgreich gelöste Laborübungen
Ermittlung der Modulnote	Klausur
Anerkannte Module	Module vergleichbaren Inhalts

Inhalte	<u>Im seminaristischen Unterricht</u> <ul style="list-style-type: none">• Modellbildung, Simulation und Analyse komplexer technischer Systeme (Informations- und Steuerungssysteme)• Vorstellung unterschiedlicher Entwicklungsmethoden: V-Modell, Unified Modeling Language (UML), Model-Based Design (MBD)• Durchgängiger Entwicklungsprozess: Von der Spezifikation über das System-Design bis zur Verifikation und Implementierung• Applikationsbeispiele aus der Mess-, Steuerungs- und Regelungstechnik
Inhalte	<u>In den Übungen</u> <ul style="list-style-type: none">• Einführung in eine Entwicklungsumgebung nach der Methode des MBD• Entwicklung signalverarbeitender Algorithmen• Simulation analoger, digitaler und gemischter Signale• Rapid Prototyping und Erzeugung von Echtzeit-Code• Implementierung und Verifikation auf Mikrocontroller-Targets
Literatur	- Tim Weilkiens: Systems Engineering mit SysML/UML; Modellierung, Analyse, Design; dpunkt.verlag - James Dabney, Thomas Harman: Mastering Simulink; Prentice Hall
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Modulnummer	ACS
Titel	Vertiefung Regelungstechnik / Advanced Control Systems
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	<p>Aufbauend auf Grundkenntnissen der klassischen Regelungstechnik werden zwei moderne Methoden der Regelungstechnik, die</p> <ul style="list-style-type: none"> • Regelungstechnik im Zustandsraum und die • Regelungstechnik mit unscharfer (Fuzzy) Logik <p>gelehrt.</p> <p>Die Regelungstechnik im Zustandsraum führt nicht die Regelgröße im Kreis zurück, sondern als Informationsgewinn alle Zustände der Strecke. Die Regelungstechnik mit Fuzzy-Logik benutzt zur Beschreibung der Strecke kein mathematisches Modell, sondern natürlichsprachlich formulierte Wenn / Dann - Entscheidungen reichen aus.</p> <p>Die Studierenden erwerben Kenntnisse über die zugrunde liegenden Regelungsphilosophien. Durch praktische Anwendung der kennengelernten analytischen und numerischen Entwurfsstrategien in rechnergestützten Übungen mit Matlab / Simulink erhalten die Studierenden die Fertigkeit, Regler der oben genannten Typen zu entwickeln und zu erproben. Die Studierenden besitzen die Kompetenz zu bewerten, ob für ein vorliegendes Regelungsproblem eine klassische, eine Zustandsraum- oder eine Fuzzy-Methode optimal einsetzbar ist.</p>
Voraussetzungen	Empfehlung : Gute Matlab-Kenntnisse
Niveaustufe	2. Studienplansemester
Lernform	Seminaristischer Unterricht mit Hausübung, Übung
Status	Wahlpflichtmodul
Häufigkeit des Angebotes	Wintersemester
Prüfungsform	Klausur. Voraussetzung zur Zulassung zur Klausur: Vorlage aller Hausübungen der Gruppe und Rücksprache „m.E.“
Ermittlung der Modulnote	Klausurnote
Anerkannte Module	Module vergleichbaren Inhalts

Inhalte	<p><u>Im seminaristischen Unterricht</u></p> <p>Regelungstechnik im Zustandsraum:</p> <ul style="list-style-type: none"> • Zustandsmodell und Strukturbilder, • Strukturell besondere Formen des Zustandsmodells (Diagonalform, Beobachter-Normalform, Regler-Normalform) • Systemeigenschaften (Steuerbarkeit und Beobachtbarkeit, Stabilität) • Regelkreise im Zustandsraum (Entwurf eines Zustandsreglers, Entwurf eines Zustandsbeobachters, Zustandsregelkreise mit Beobachter, geführte und gestörte Zustandsregelkreise) • Zeitdiskrete Systeme im Zustandsraum. <p>Regelungstechnik mit unscharfer (Fuzzy-) Logik:</p> <ul style="list-style-type: none"> • Grundlagen und Grundbegriffe (Unscharfe Logik, Grundbegriffe (Zugehörigkeitsfunktion, Fuzzy-Set, Linguistische Variable), Operationen mit Fuzzy-Sets (UND-, ODER-Operatoren, Komplement- und andere Operatoren) • Klassische und Fuzzy-Ansätze für eine Reglerentwurf • Grundlagen des Fuzzy-Regler-Entwurf • Aktionsfolgen eines Fuzzy-Reglers (Fuzzifizierung, Aggregation, Implikation, Akkumulation, Defuzzifizierung) • Ein praktisches Beispiel (Wahl der Eingangs- und Ausgangsgrößen, Festlegung der linguistischen Terme, Festlegung der Produktionsregeln, der Aktionsablauf im Fuzzy-Regler) • Einführung von Dynamikanteilen in den Regler, bzw. Erweiterung der Anzahl der Eingänge des Regler (Festlegung der linguistischen Terme, Festlegung der Produktionsregeln, der Aktionsablauf im Fuzzy-Regler mit zwei Eingängen). <p><u>In den Übungen</u></p> <ul style="list-style-type: none"> • Entwurf und Simulation von Zustandsregelkreisen unter Matlab-Simulink. • Entwurf und Simulation von Fuzzy-Reglern und –Regelkreisen (Aufbau und Anwendung der Fuzzy-Logic-Toolbox von Matlab/Simulink).
Literatur	<p>H. Unbehauen: „Regelungstechnik II“, Friedr. Vieweg & Sohn Braunschweig/Wiesbaden Z. Kovacic, S. Bogdan: „Fuzzy controller design“, CRC/Taylor Francis, Boca Raton, FL</p>
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Modulnummer	AMS
Titel	Autonome Mobile Systeme / Autonomous Mobile Systems
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	<p>Die Studierenden werden mit den vielseitigen Aspekten mobiler Roboter vertraut gemacht. Nach einem Anwendungsüberblick lernen die Studenten den grundlegende Aufbau dieser komplexen Systeme bestehend aus den Modulen Antrieb und Bewegung, Rechner mit Steuersoftware, Energieversorgung und Sensorik kennen. Ein wesentlicher Schwerpunkt liegt in der Wahrnehmung (Perception) und internen Darstellung der Umgebung des Roboters (Mapping). Hierauf aufbauend werden bekannte Navigationsverfahren behandelt, die den Roboter in die Lage versetzen, sich selbständig in unbekanntem Umgebungen zu orientieren bzw. ein beliebiges Ziel anzusteuern. Vertiefte Kenntnisse werden insbesondere beim Thema Lokalisierung vermittelt, wobei etablierte Ansätze wie Scan Matching, Kalman-Filterung oder Markov-Lokalisierung einschließlich aktueller Erweiterungen (Monte-Carlo-Lokalisierung, SLAM-Problem) diskutiert werden.</p> <p>Die Studierenden erwerben die Kompetenz, mobile Roboter für vorgegebene Aufgaben unter Einsatz von komplexen Navigationsalgorithmen zu entwickeln.</p>
Voraussetzungen	Empfehlung: Kenntnisse, wie sie in den Modulen „Mathematik“, „Programmierung“, „Systemtheorie“, „Regelungstechnik“ des Bachelor-Studiengangs „Technische Informatik“ und im Wahlpflichtmodul „Adaptive Filter“ vermittelt werden.
Niveaustufe	2. Studienplansemester
Lernform	Seminaristischer Unterricht und Laborübungen unter Einsatz eines Roboter-Experimentiersystems
Status	Pflichtmodul
Häufigkeit des Angebotes	Wintersemester
Prüfungsform	Mündliche Prüfung und Anerkennung der Laborübungen
Ermittlung der Modulnote	Prüfungsnote
Anerkannte Module	Module vergleichbaren Inhalts

Modulnummer	RQE
Titel	Anforderungsmanagement / Requirements Engineering
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	<p>Die Studierenden besitzen die Fertigkeit, bei der Entwicklung eines Systems die Anforderungen von Kunden zu bewerten, zu erfassen, korrekt zu modellieren und an folgende Entwicklungsschritte anzufügen. Sie verfügen über Fähigkeiten, diese Anforderungen im Zuge einer Systementwicklung zu verfeinern, auf die einzelnen Entwicklungsschritte abzubilden und aktuell zu halten. Sie haben die Kompetenz erworben, Problemen, die im Umgang mit Kunden und Entwicklern häufig auftreten, zu erkennen und systematisch zu lösen. Sie kennen Techniken zum Test und Nachweis der Erfüllung der Anforderungen, beherrschen die Nutzung eines gängigen RE-Werkzeuges und können die Leistungsfähigkeit von Werkzeugen am Markt beurteilen..</p> <p>Die Studierenden wissen, wie die Attribute der Anforderungen zur Anforderungsverfolgung und zur Projektplanung und –Verfolgung genutzt werden können.</p> <p>Damit erreichen Sie wichtige Kompetenzen zur selbständigen inhaltlichen Entwicklung und Steuerung von Projekten.</p>
Voraussetzungen	keine
Niveaustufe	2. Studienplansemester
Lernform	Seminaristischer Unterricht und Übungen am Digitalrechner
Status	Pflichtmodul
Häufigkeit des Angebotes	Wintersemester
Prüfungsform	Klausur und Bewertung der Laborübungen
Ermittlung der Modulnote	70% Klausurnote, 30% Bewertung der Laborübungen
Anerkannte Module	Module vergleichbaren Inhalts
Inhalte	<p><u>Im seminaristischen Unterricht</u></p> <ul style="list-style-type: none"> • Anforderungen und Vorgehen bei der Arbeit mit ihnen • Erfassung, Klassifizierung und Analyse der Anforderungen • Methoden und Techniken der Modellierung von Anforderungen • Validierung von Anforderungen • Verwaltung von Anforderungen • Metriken auf Basis von Anforderungen • Überblick über Werkzeuge zum Requirements-Management <p><u>In den Übungen</u></p> <p>Die Nutzung der in dem Seminaristischen Unterricht erarbeiteten Kenntnisse wird in Projektaufgaben in Teams von 4 Studierenden umgesetzt.</p>

Literatur	C. Ebert: Systematisches Requirements Management, d.punkt Verlag G. Kotonya, I. Sommerville: Requirements Engineering, Wiley
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Modulnummer	CES
Titel	Konfigurierbare eingebettete Systeme / Configurable Embedded Systems
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU + 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	Die Studierenden besitzen die nötigen Kenntnisse, um eine vorgegebene Aufgabenstellung für ein eingebettetes Design zu analysieren und mit modernen Methoden des Plattform-basierten Entwurfs eine an 'Time-to-Market'-Vorgaben orientierte Lösung selbstständig zu entwickeln. Sie können auf dem Desk-Top komplette Anwendungs-Systeme bestehend aus Hard- und Software als System-on-a-Chip (SoC) entwickeln und verifizieren. Im Zusammenhang mit anderen Veranstaltungen haben sie gelernt, alternative Realisierungsverfahren zu bewerten und zu vergleichen, um die für die jeweiligen Rahmenbedingungen optimale Methodik einzusetzen. Hieraus entwickeln sich Kompetenzen, welche einerseits fachspezifischer Art sind und auf der Kenntnis des Zusammenspiels komplexer Werkzeuge beruhen, andererseits können sie eine fachgebiets-übergreifende Entwicklung und Simulation kompletter Anwendungssysteme in Team-Arbeit durchführen.
Voraussetzungen	keine
Niveaustufe	2. Studienplansemester
Lernform	Seminaristischer Unterricht und Laborübungen in Form eines Kleinprojekts
Status	Pflichtmodul
Häufigkeit des Angebotes	Wintersemester
Prüfungsform	Klausur und Bewertung der Laborübungen
Ermittlung der Modulnote	70% Klausurnote, 30% Bewertung der Laborübungen
Anerkannte Module	Module vergleichbaren Inhalts

<p>Inhalte</p>	<p><u>Im seminaristischen Unterricht</u></p> <ul style="list-style-type: none"> • Grundlagen der Hardware-Strukturen für konfigurierbare eingebettete Systeme • Prozessor-Architekturen und Bus-Systeme für den SoC-Entwurf • Entwurfsmethoden (vom Timing Driven Design über den Block Based Design zum Platform Based Design) • Einsatz und Entwurf virtueller Hardware-Komponenten • Simulations- und Debugverfahren • Hardware-/Software Co-Design und Software-Konzepte für den SoC-Entwurf • Reprogrammierung von Hardware • Spezielle Anforderungen an Operating System, Compilation System und Runtime System für rekonfigurierbare Hardware in eingebetteten Systemen • Internet Reconfigurable Logic (IRL) <p><u>In der Übung</u></p> <ul style="list-style-type: none"> • Benutzung des 'Embedded Development Kit' von XILINX • Zusammenstellen von Prozessor-Systemen mit dem XILINX Platform Studio • Entwicklung eigener virtueller Komponenten in VHDL • Simulation durch funktionale Bus-Modellierung • Softwareentwicklung als ISR bzw. uClinux Anwendung • Hardware-Debugging mit ChipScope • Projekt zur Realisierung einer SoC-Anwendung
<p>Literatur</p>	<p>- Frank Vahid and Tony Givargis: Embedded System Design: A Unified Hardware/Software Introduction, Wiley - Henry Chang, Larry Cooke, Merrill Hunt, Grant Martin, Andrew McNelly, Lee todd: Surviving the SOC Revolution: A Guide to Platform-Based Design, Kluwer Academic Publishers</p>
<p>Weitere Hinweise</p>	<p>Dieses Modul wird auf Deutsch angeboten.</p>

Modulnummer	CTE
Titel	Modul mit aktuellen Inhalten zu Embedded Systems / Contemporary Topics in Embedded Systems
Credits	5 Cr
Präsenzzeit	4 SWS (2 SWS SU+ 2 SWS Ü)
Workload	SU: ~ 36 h Ü: ~ 36 h Selbstlernzeit: ~ 78 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	Die Studierenden sollen <u>aktuelle</u> Problemstellungen von eingebetteten Systemen kennen- und bearbeiten lernen
Voraussetzungen	keine
Niveaustufe	2. Studienplansemester
Lernform	Seminaristischer Unterricht mit Hausübung
Status	Wahlpflichtmodul
Häufigkeit des Angebotes	Wintersemester
Prüfungsform	Klausur. Voraussetzung für die Zulassung zur Klausur: Vorlage aller Hausübungen der Gruppe und Rücksprache „m.E.“
Ermittlung der Modulnote	Klausurnote
Anerkannte Module	Module vergleichbaren Inhalts
Inhalte	Die Inhalte werden semesterweise festgelegt. Sie orientieren sich an aktuellen Entwicklungstendenzen eingebetteter Systeme.
Literatur	Je nach ausgewählten Themen
Weitere Hinweise	Dieses Modul wird auf Deutsch/Englisch angeboten.

Modulnummer	MTH
Titel	Master-Arbeit mit integriertem Kolloquium/ Master Thesis (Abschlussarbeit gemäß RPO III)
Credits	25 Cr
Präsenzzeit	1 SWS Teilnahme am Masterkolloquium und Betreuungsgespräche
Workload	Präsenzzeit im Kolloquium: ca. 18 h Selbstlernzeit: ~ 730 h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	Die Studierenden können eine komplexe, umfangreiche Aufgabenstellung aus dem Gebiet Embedded Systems mit wissenschaftlichen Methoden in einem vorgegebenen Zeitrahmen selbständig bearbeiten, lösen und dokumentieren. Sie können Fachinhalte Ihrer Arbeit präsentieren und in kritischer Diskussion in einem Fachkolloquium verteidigen.
Voraussetzungen	Zulassung gemäß Prüfungsordnung des Master-Studiengangs Embedded Systems
Niveaustufe	3. Studienplansemester
Lernform	Betreute, selbständig durchzuführende wissenschaftliche Arbeit mit integriertem Kolloquium
Status	Pflichtmodul
Häufigkeit des Angebotes	jedes Semester
Prüfungsform	Beurteilung der Lösungsgüte der Aufgabenstellung und der Dokumentation durch die Prüfungskommission.
Ermittlung der Modulnote	Ergebnis der Beurteilung der Lösungsgüte und der Dokumentation der Master-Arbeit durch die Prüfungskommission
Anerkannte Module	keine
Inhalte	Problemstellungen aus dem Gebiet Embedded Systems, wie sie von entsprechenden Industrieunternehmen, Forschungseinrichtungen, Partnerhochschulen und Lehrkräften der TFH formuliert werden. Auch Aufgabenstellungen, die von den Studierenden formuliert werden, können mit Einverständnis des betreuenden Hochschullehrers bearbeitet werden.
Literatur	-
Weitere Hinweise	Dieses Modul wird auf Deutsch angeboten. Im Einvernehmen mit dem Betreuer kann die Master-Arbeit in englischer Sprache verfasst werden. Die Masterarbeit muss eine Zusammenfassung in deutscher und englischer Sprache enthalten. Das Modul beinhaltet die Teilnahme an einem Masterkolloquium. Dauer: 5 Monate

Modulnummer	OEX
Titel	Mündliche Abschlussprüfung(gemäß RPO III) / Oral Examination
Credits	5 Cr
Präsenzzeit	45 – 60 Minuten Prüfung
Workload	150h
Lerngebiet	Fachspezifische Vertiefung
Lernziele / Kompetenzen	Durch die Prüfung wird festgestellt, ob der Prüfling gesichertes Wissen in den Fachgebieten, denen diese Arbeit thematisch zugeordnet ist, besitzt und fähig ist, die Ergebnisse der Master-Arbeit selbstständig zu begründen sowie die Grundlagen des gesamten Studiengbietes klar und verständlich zu erläutern. Der Prüfling zeigt seine Kommunikationskompetenz in der verdichteten Präsentation eines komplexen Sachverhalts.
Voraussetzungen	Abschluss aller Module einschließlich der Master-Arbeit
Niveaustufe	3. Studienplansemester
Lernform	Selbständige Vorbereitung auf die Abschlusspräsentation und -Prüfung
Status	Pflichtmodul
Häufigkeit des Angebotes	jedes Semester
Prüfungsform	Mündliche Abschlussprüfung vor der Prüfungskommission
Ermittlung der Modulnote	Benotung der Präsentation und der Befragung durch die Prüfungskommission
Anerkannte Module	keine
Inhalte	Die mündliche Abschlussprüfung orientiert sich schwerpunktmäßig an den Fachgebieten der Master-Arbeit. Grundlagen der anderen Module können anteilig Prüfungsthema sein. Die 15-Minütige Präsentation gibt einen kurzen Überblick über die Aufgabenstellung und die erreichten Ergebnisse..
Literatur	--
Weitere Hinweise	Nach Vereinbarung zwischen Prüfling und Prüfungskommission kann die Präsentation auch auf Englisch erfolgen.